

CASIO DT-930 C Language SDK GUI Version

アプリケーション開発ガイド

DT-930 C Language SDK GUI Version を使用した、DT-930 のアプリケーションプログラム開発の概要（プログラムの開発、ダウンロード等）を、開発の流れにしたがって説明します。



ご注意

- このソフトウェアおよびマニュアルの、一部または全部を無断で使用、複製することはできません。
- このソフトウェアおよびマニュアルは、本製品の使用許諾契約書のもとでのみ使用することができます。
- このソフトウェアおよびマニュアルを運用した結果の影響については、一切の責任を負いかねますのでご了承ください。
- このソフトウェアの仕様、およびマニュアルに記載されている事柄は、将来予告なしに変更することがあります。
- このマニュアルの著作権はカシオ計算機株式会社に帰属します。
- 本書中に含まれている画面表示は、実際の画面とは若干異なる場合があります。予めご了承ください。

© 2007 カシオ計算機株式会社

Microsoft, MS, ActiveSync, Active Desktop, Outlook, Windows, Windows NT, および Windows ロゴは、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。Microsoft 社の製品は、OEM 各社に、Microsoft Corporation の 100%出資子会社である Microsoft Licensing, Inc.によりライセンス供与されています。

目次

1.	はじめに	1
1.1.	概要	1
1.2.	開発環境	2
2.	アプリケーションの開発方法	8
2.1.	アプリケーションの開発準備	9
2.1.1.	周辺機器の用意	9
2.1.2.	ファイル転送ツールの登録	11
2.1.3.	拡張機能の登録	13
2.2.	アプリケーションの作成	16
2.2.1.	eMbedded Visual C++ 4.0 の起動	16
2.2.2.	アプリケーションウィザード	16
2.2.3.	アプリケーションの作成/編集(新規アプリケーション)	22
2.2.4.	アプリケーションの作成/編集(新規/既存共通)	24
2.2.5.	アプリケーションの作成/編集(既存アプリケーション)	25
2.3.	アプリケーションのビルド	26
2.3.1.	ビルド方法	26
2.3.2.	生成するオブジェクトの種類	28
2.4.	アプリケーションのデバッグ	29
2.4.1.	デバッグ基本操作	29
2.4.2.	エミュレータデバッグ	31
2.5.	アプリケーションのインストール	34
3.	エミュレータの使用方法	35
3.1.	DT-930 エミュレータ	35
3.1.1.	キー入力機能	35
3.1.2.	表示機能	35
3.1.3.	バーコード入力機能	35
3.1.4.	サウンド機能	35
3.1.5.	インジケーション機能	36
3.2.	I/O シミュレータ	37
3.2.1.	入力バーコードの登録	37
3.2.2.	ローバッテリー発生/IOBOX 接続動作	39
3.3.	実機とエミュレータの動作の違い	40
3.3.1.	デバッグ	40
3.3.2.	ドライブレター	40
3.3.3.	通信機能	41
3.3.4.	システムファイル	41
3.3.5.	副電池	41
3.3.6.	バックライト/コントラスト	41
3.3.7.	DT-700 互換表示モード	41
3.3.8.	ユーザフォント/外字	41
3.3.9.	関数インタフェース	42
4.	その他の機能	49
4.1.	フォントコンバータ	49
4.1.1.	機能	51

4.1.2.	画面構成	53
4.1.3.	使用方法	57
4.1.4.	注意事項	65
4.2.	LMWIN	66
4.2.1.	機能	66
4.2.2.	使用方法	66
4.3.	MultiDrop	73
4.3.1.	機能	73
4.3.2.	使用方法	73

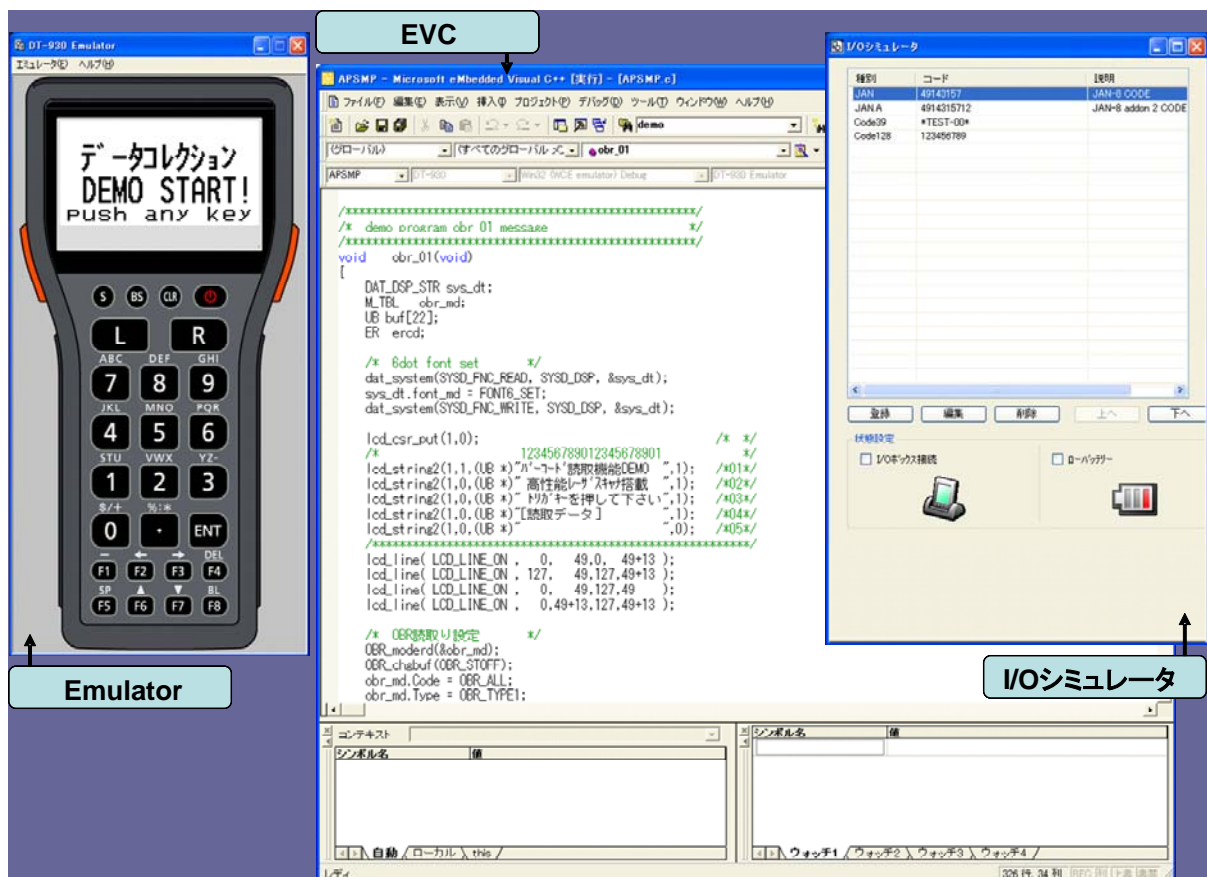
1. はじめに

このたびは、CASIO DT-930 C Language SDK GUI Version をお買い上げありがとうございます。
本書は、DT-930 のアプリケーション開発キット「DT-930 C Language SDK GUI Version」を使用して、アプリケーションの開発をするための手順を説明したものです。

1.1. 概要

OSとして μ ITRONを搭載するDT-900シリーズは、これまでDosプロンプト上でコマンドラインを使用したCUI環境でアプリケーション開発を行ってきましたが、本SDKでは、Microsoft eMbedded Visual C++4.0 (eVC)を使用することにより、下図のように、ビジュアルなマンマシンを持ったGUI環境での開発が可能となります。

さらに、作成したアプリケーションをPC上でデバッグ可能な、「Emulator」と「I/Oシミュレータ」を搭載しており、これにより、アプリケーションのデバッグを効率良く行うことが可能となります。



1.2. 開発環境

アプリケーションを開発するためには、本 SDK の他に、下記の環境が必要となります。

開発環境	説明
ハードウェア(PC)	IBM/PC AT 互換機
OS	下記のいずれか • Winsows2000 Pro SP4 • WindowsXP Pro SP2 • Windows2000 Server SP4 • Windows Server2003 SP1 • Windows Vista
ソフトウェア	• Microsoft eMbedded Visual C++ 4.0 • Microsoft eMbedded Visual C++ 4.0 SP4 通信キット(下記のいずれか) • LMWIN Version6.13 以降 • Multi DROP Version2.00 以降
ターゲットデバイス	DT-930

データ通信用の IOBOX やポケットプリンタ等のオプションデバイスに関しては、必要に応じて別途ご用意ください。

また、DT-930 エミュレータを起動するには、少なくともひとつの仮想ネットワーク環境が必要です。使用されるコンピュータからネットワークをアクセスできない場合は、Microsoft Loopback Adapter をインストールする必要があります。以下に、Microsoft Loopback Adapter のインストール方法を説明しますが、更に詳しい情報は MSDN ライブラリで"loopback adapter"をキーに検索してください。

Microsoft Loopback Adapter のインストール方法

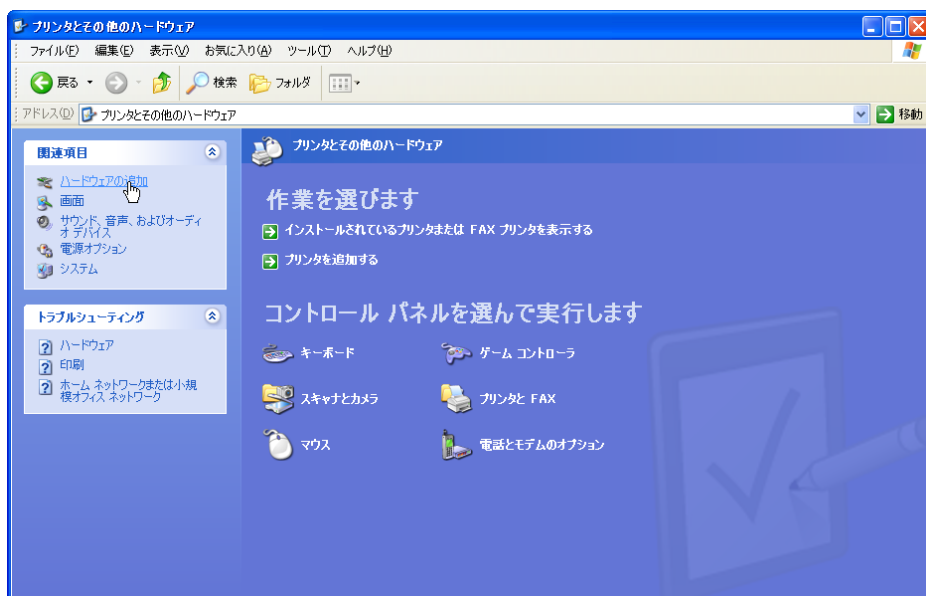
DT-930 エミュレータはネットワーク環境で動作するように設計しています。しかし、出先でのデモンストレーションやデバッグなど、ネットワークがない環境でもエミュレータを動作させるには仮想ネットワークを用意する必要があります。

仮想ネットワークの構築は簡単で、マイクロソフト製の仮想 LAN ボードである Loopback Adapter を以下の手順に従ってインストールするだけです。なお、以下の手順は日本語版 Windows XP Professional で説明していますので、それ以外の環境では適宜読み替えてください。

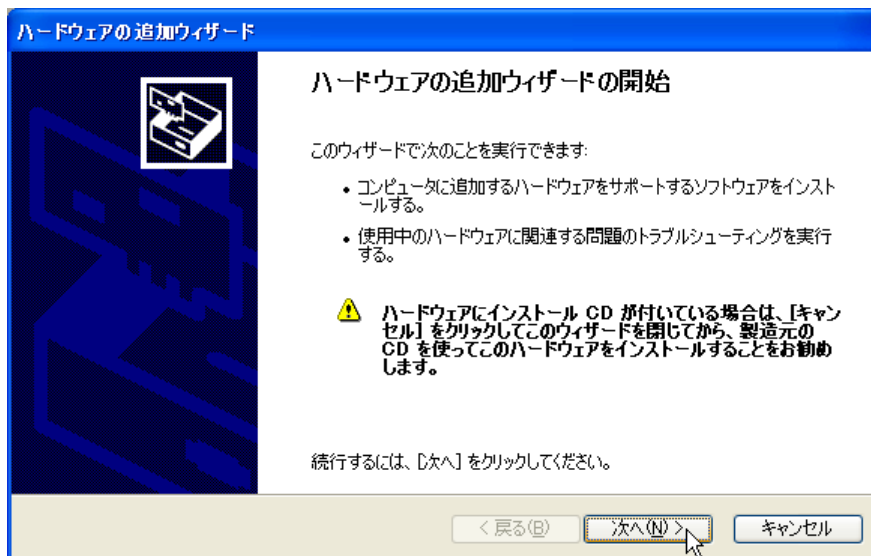
1. コントロールパネルを開き、「プリンタとその他のハードウェア」をクリックします。



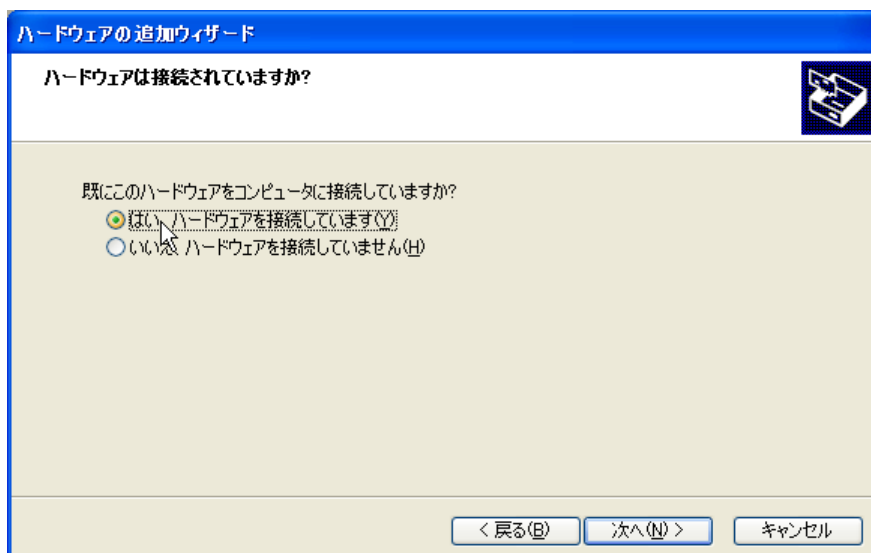
2. 「ハードウェアの追加」をクリックします。



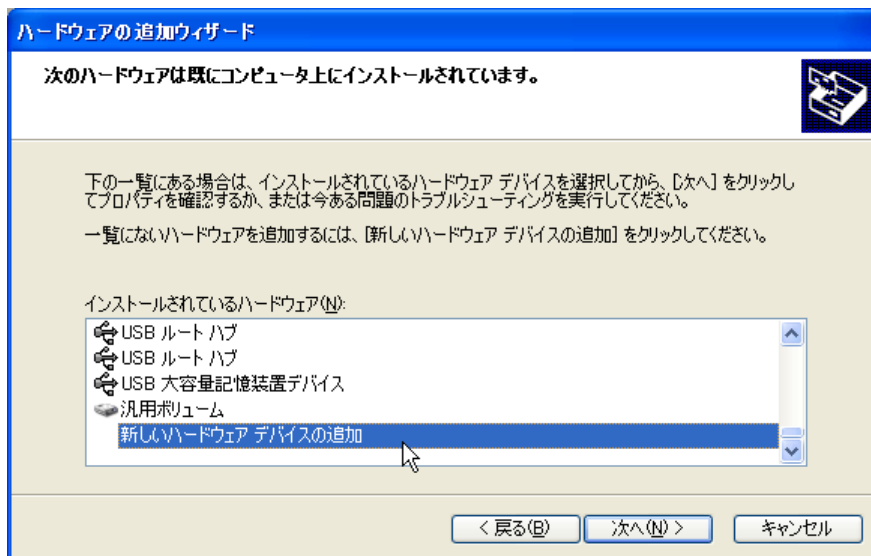
3. ハードウェアの追加ウィザードが開始したら、「次へ(N) >」をクリックします。



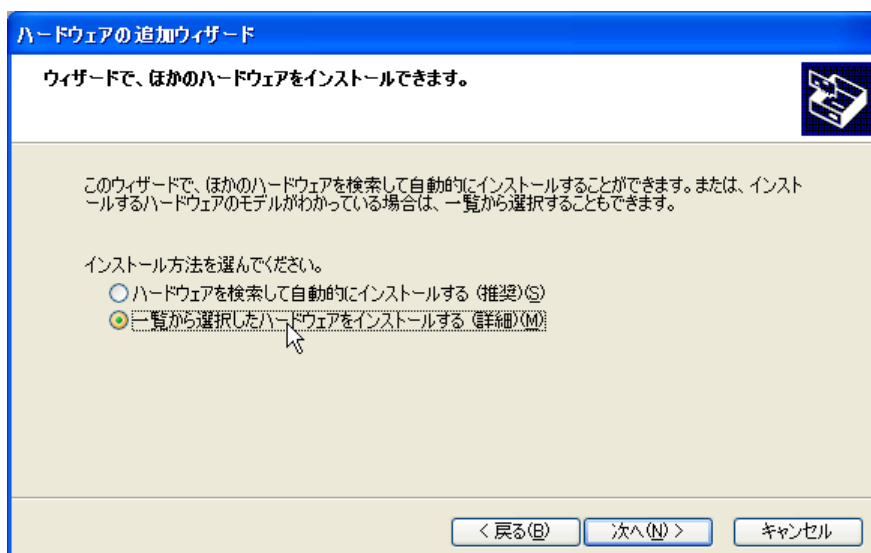
4. 「はい、ハードウェアを接続しています(Y)」を選択して「次へ(N) >」をクリックします。



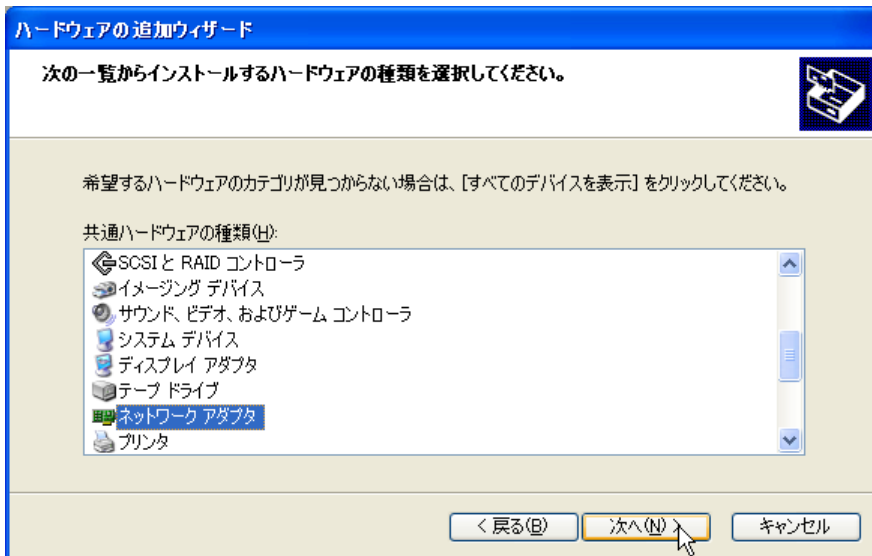
5. リストの一番下にある「**新しいハードウェアの追加**」を選択して「**次へ(N) >**」をクリックします。



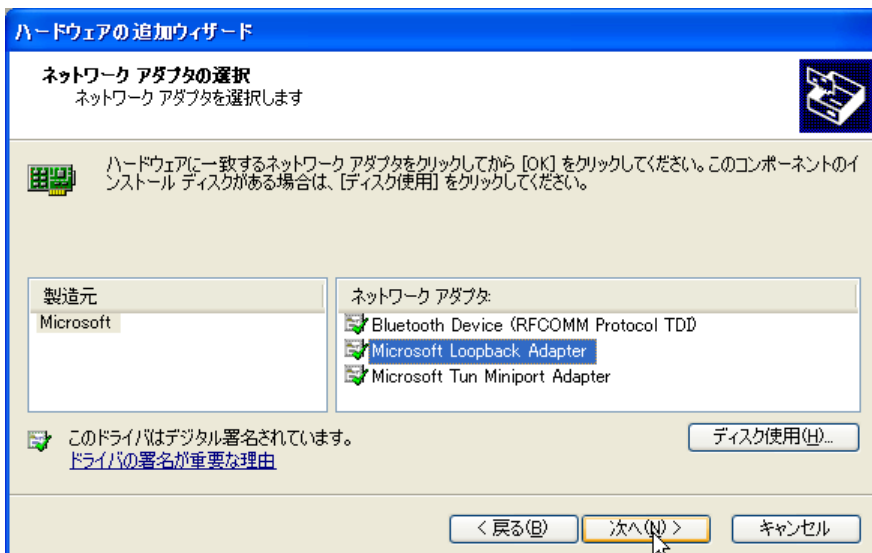
6. 「**一覧から選択したハードウェアをインストールする (詳細)(M)**」を選択して「**次へ(N) >**」をクリックします。



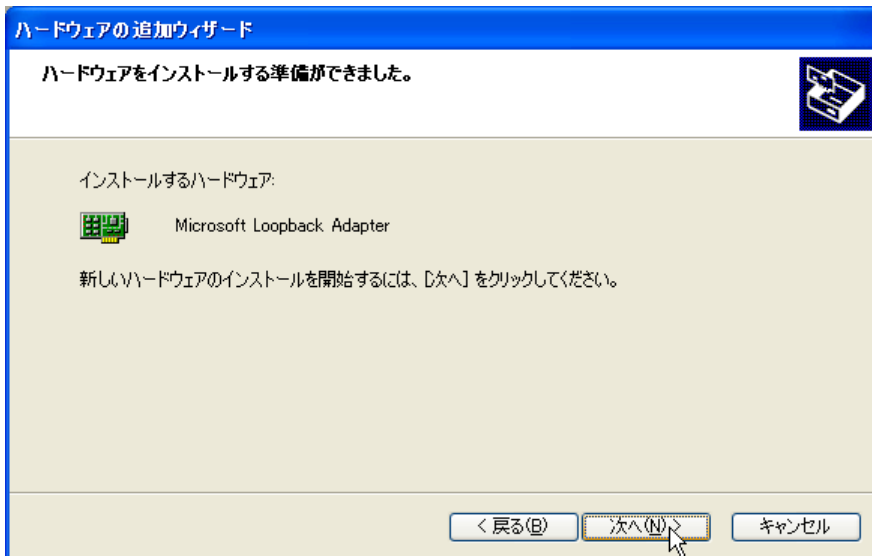
7. 一覧から「**ネットワークアダプタ**」を選択して「**次へ(N) >**」をクリックします。



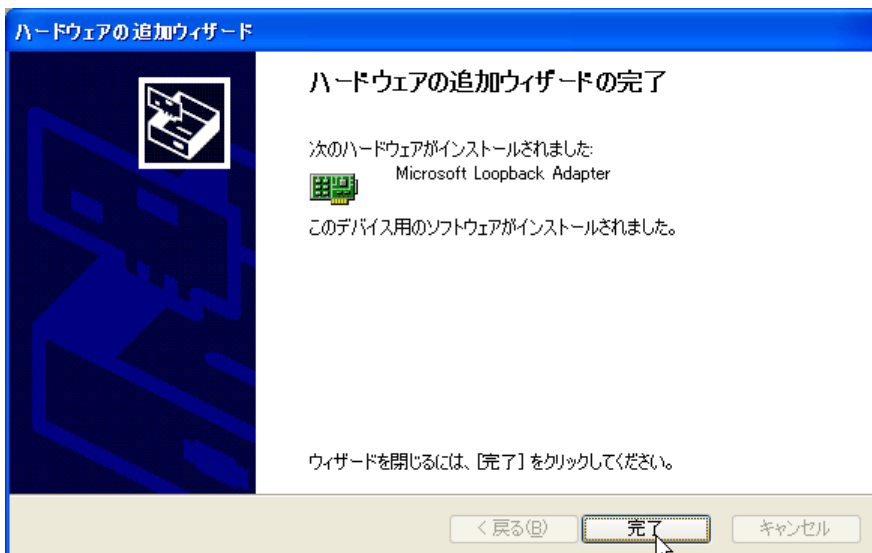
8. 一覧から「**Microsoft Loopback Adapter**」を選択して「**次へ(N) >**」をクリックします。



9. 以下のダイアログが出たら「次へ(N) >」をクリックします。



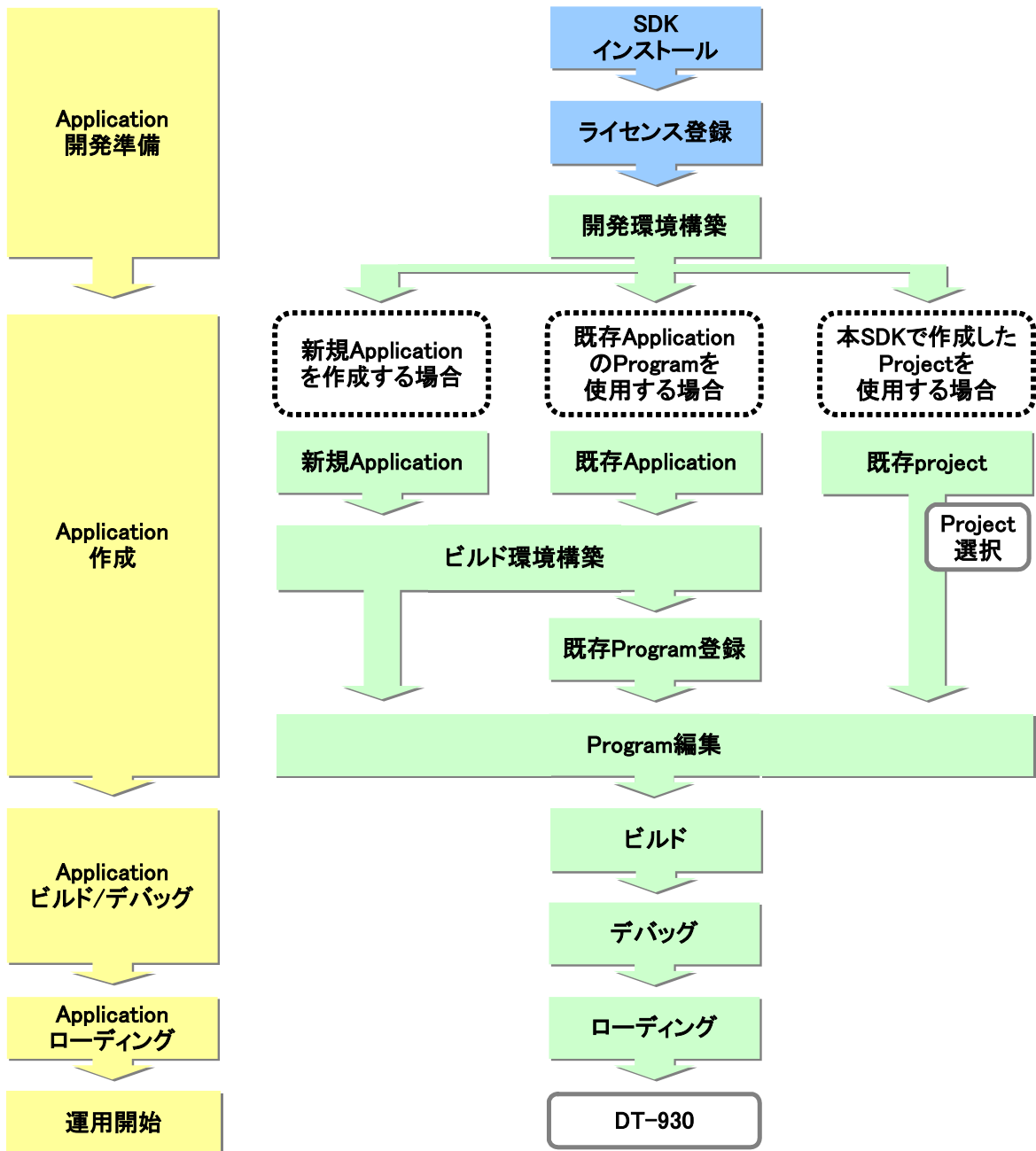
10. 以上で、Microsoft Loopback Adapter のインストールは完了です。



2. アプリケーションの開発方法

アプリケーションの開発は、下図のような流れで行います。

eVC のマンーマシンインタフェースを使用した開発となりますが、eVC に対する詳しい知識がなくても、簡単にアプリケーションの開発ができることが特徴です。



2.1. アプリケーションの開発準備

本 SDK をインストール後、ライセンス登録を行い、開発環境(周辺機器等)を構築することにより、アプリケーションの開発準備が整います。

※ インストールおよびライセンス登録の方法は、「ファーストステップガイド」を参照してください。

2.1.1. 周辺機器の用意

作成したアプリケーションを DT-930 にローディングするための IOBOX や、運用で使用するポケットプリンタ等のオプション機器を用意してください。

本体	説明
DT-930M50	レーザーキャナ ベントモデル
DT-930M51	レーザーキャナ ストレートモデル
IOBOX	説明
HA-E60IO	USB クレードル
DT-960IO	ベーシック I/O ボックス
DT-964IOA/DT-964IO	サテライト I/O ボックス

充電器	説明
DT969CHG	リチウムイオン充電機専用充電器
リチウムイオン充電機パック	説明
DT-923LIB	リチウムイオン充電機パック

通信ケーブル	説明
DT-881RSC	DT-964IO(A)のRS-232Cインターフェースとモデムを接続する時に使用します。
DT-882RSC	DT-964IO(A)のRS-232Cインターフェースとパソコンを接続する時に使用します。 (RS-232Cコネクタ:D-sub25ピン メス)
DT-883RSC	DT-964IO(A)のRS-232Cインターフェースとパソコンを接続する時に使用します。 (RS-232Cコネクタ:D-sub25ピン オス)
DT-887AX	DT-964IO(A)のRS-232Cインターフェースとパソコンを接続する時に使用します。 (RS-232Cコネクタ:D-sub9ピン オス)
DT-888RSC	DT-964IO(A)を複数台連鎖接続する時に使用します。
DT-782RSC	DT-960IOのRS-232Cインターフェースとパソコンを接続する時に使用します。 (RS-232Cコネクタ:D-sub25ピン メス)
DT-783RSC	DT-960IOのRS-232Cインターフェースとパソコンを接続する時に使用します。 (RS-232Cコネクタ:D-sub25ピン オス)
DT-787AX	DT-960IOのRS-232Cインターフェースとパソコンを接続する時に使用します。 (RS-232Cコネクタ:D-sub9ピン オス)
DT-788RSC	DT-960IOを複数台連鎖接続する時に使用します。
DT-810PR	赤外線インターフェース ポケットプリンター

ファイル転送ツール	説明
DT-UD7WIN32	通信キット:マルチドロップ (対応 I/O ボックス:DT-960IO)
DT-UD8WIN32	通信キット:LMWIN (対応 I/O ボックス:HA-E60IO / DT-964IOA)

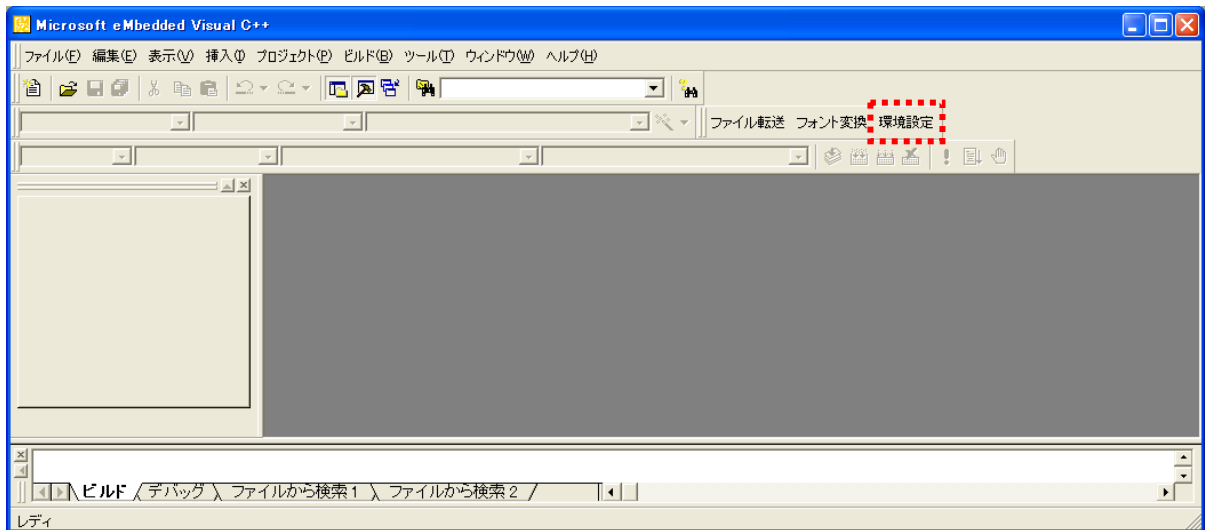
※ 各オプションの使用方法に関しては、各々のマニュアルを参照してください。

2.1.2. ファイル転送ツールの登録

IOBOX を使用して、作成したアプリケーションを DT-930 にローディングするには、ファイル転送ツールの登録が必要です。下記の手順で、登録しておくことにより、本 SDK の開発環境から直接、ファイル転送ツールを起動することが可能です。

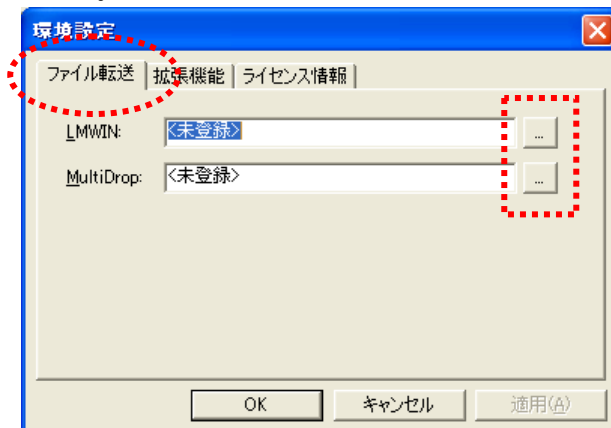
(1) 環境設定

本 SDK をインストールすることにより表示されるツールバーの「**環境設定**」をクリックします。



(2) ファイル転送ツールの選択

LMWIN または、MultiDrop の登録画面を表示したら、登録するツールの右側にあるボタンをクリックしてください。

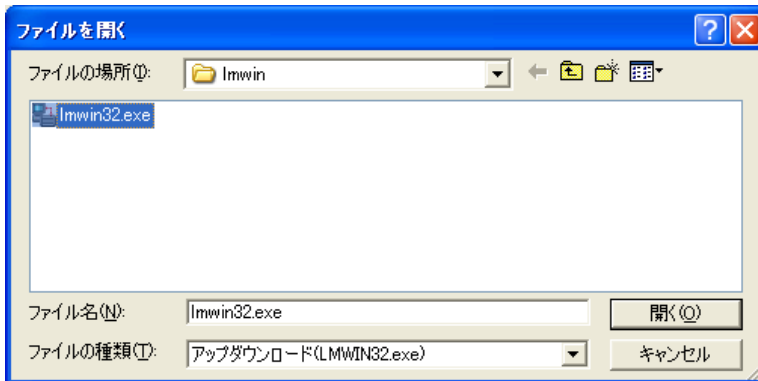


(3) ファイル転送ツールの登録

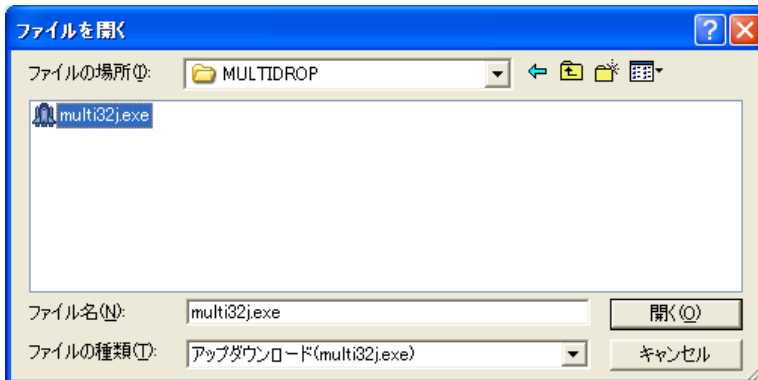
ファイル転送ツールをインストールしているフォルダを指定すると、選択したファイル転送ツールを表示します。

ツールを選択し「OK」をクリックすると登録が完了します。

- LMWIN を利用する場合

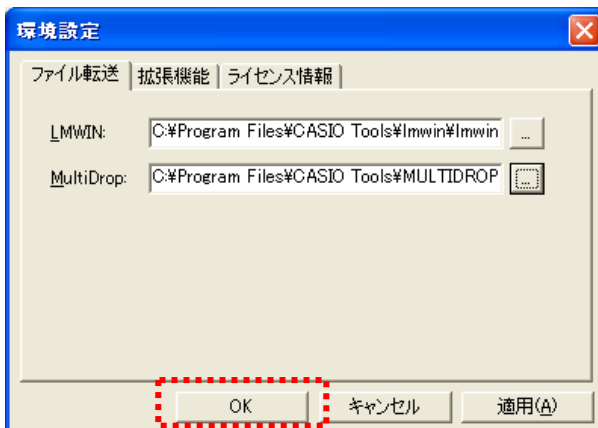


- MultiDrop を利用する場合



(4) 登録内容の確認

登録内容を確認し、「OK」ボタンをクリックしてください。



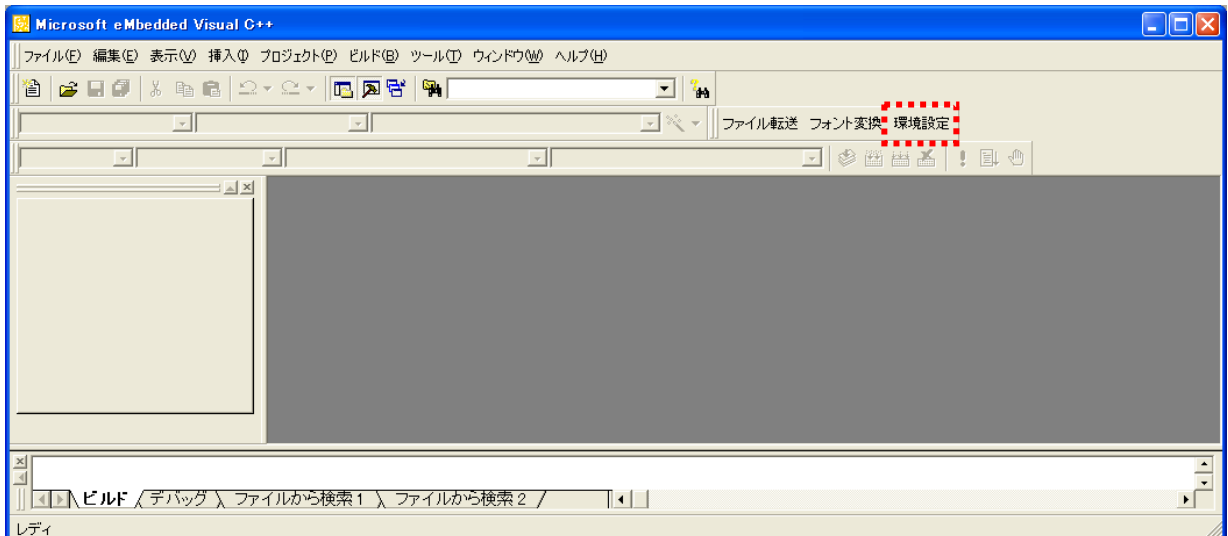
※ 2つ登録する必要はありません。ご利用になるファイル転送ツールのみを登録してください。

2.1.3. 拡張機能の登録

DT-930 は拡張機能として OBJ や LIB を提供しており、アプリケーションにリンクすることによって、様々な機能拡張が可能となります。以下の手順で、登録しておくことにより、本 SDK の開発環境に、拡張機能を組み込むことが可能です。

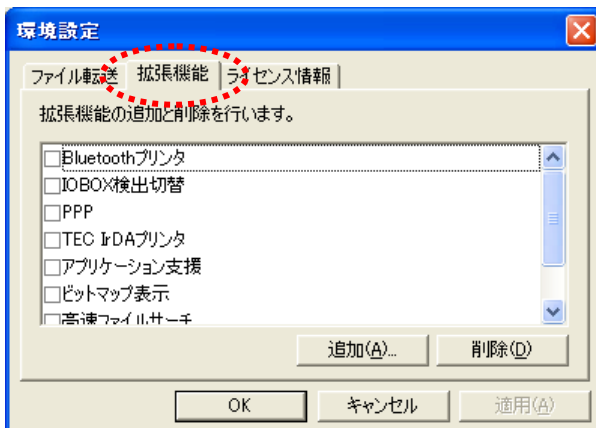
(1) 環境設定

本 SDK をインストールすることにより表示されるツールバーの「**環境設定**」をクリックします。



(2) 拡張機能の追加/削除画面

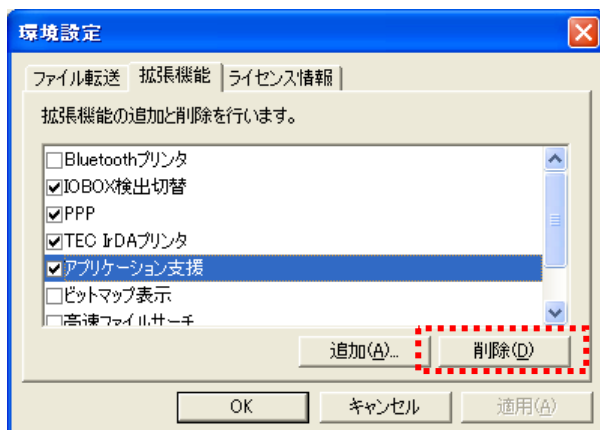
「**拡張機能**」のタブをクリックすると、拡張機能の追加/削除画面を表示します。



(3) 拡張機能の削除

削除したい拡張機能にチェックを入れ、「**削除**」ボタンをクリックすると、登録している拡張機能をすべて削除します。

この操作を行うと、登録している拡張機能のファイルと登録情報のすべてを PC より削除します。



(4) 拡張機能の登録

「**追加**」ボタンをクリックし、登録する拡張機能を格納しているフォルダを指定後、拡張機能の定義ファイル (xxxxxxx.exm) を選択してください。「**開く**」ボタンをクリックすると、選択した拡張機能を登録します。



【拡張機能定義ファイルの書式】

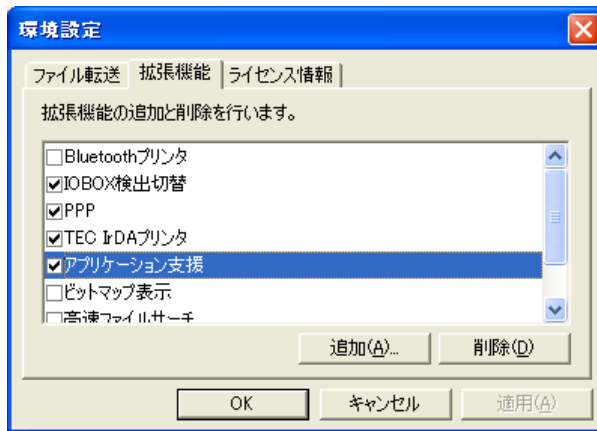
例: ADD_FUNC.exm

[MODULE]	: 固定
Desc="拡張機能"	: 一覧表示する際の表示内容を記述してください
Header=ADD_FUNC.H	: 登録する Header ファイル名を記述してください
Link=ADD_FUNC.LIB	: 登録する LIB または OBJ ファイル名を記述してください

※ 登録時は、必ず同一フォルダ内にすべての拡張機能ファイル(拡張機能定義ファイルも含めて)を入れてください。

(5) 拡張機能の登録確認

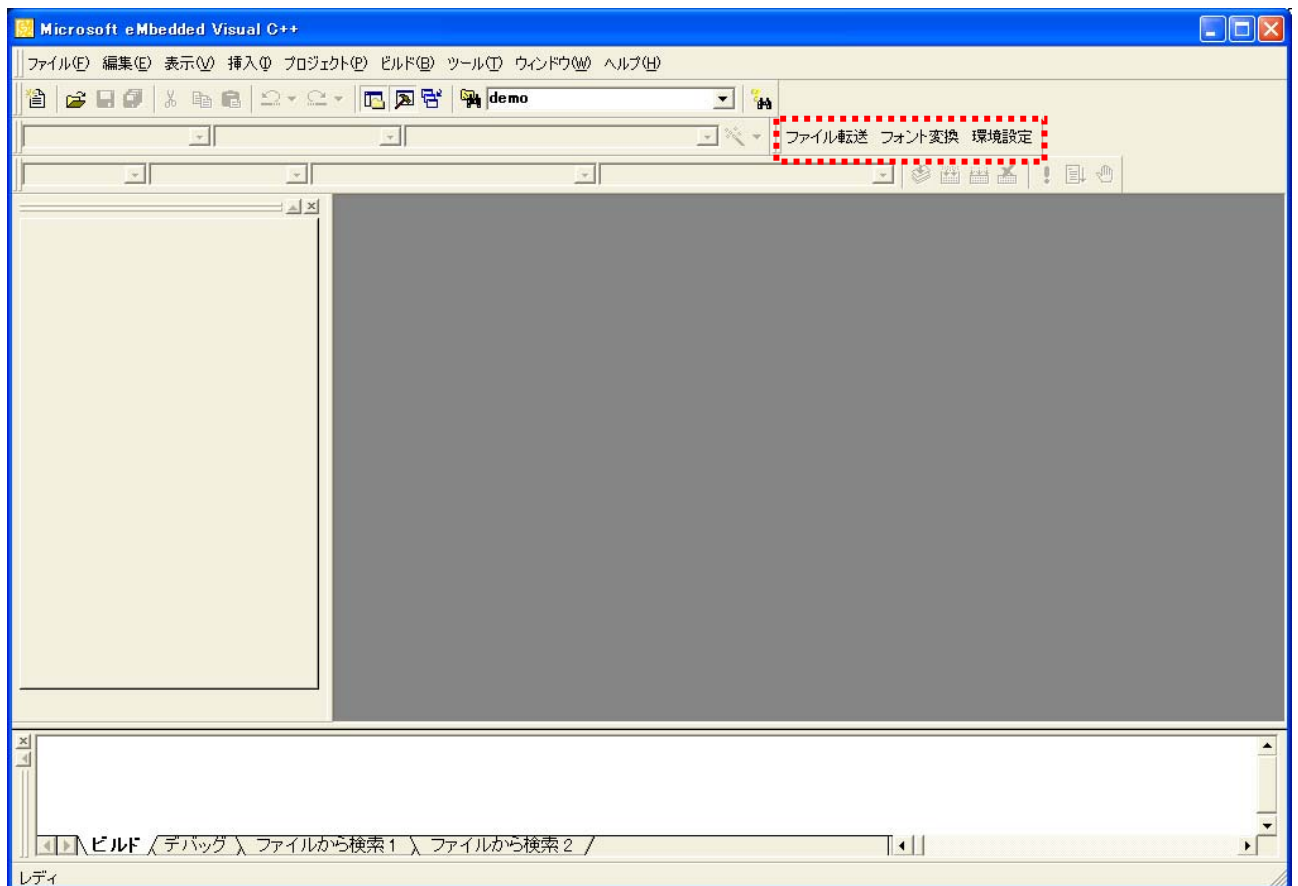
登録が完了すると、追加した拡張機能を一覧に表示します。



2.2. アプリケーションの作成

2.2.1. eMbedded Visual C++ 4.0 の起動

本 SDK をインストールすると、「ファイル転送」「フォント変換」「環境設定」のツールバーが追加されます。本 SDK を使用して作成したアプリケーションを使う場合は、そのワークスペースを開いてください。これは、通常の eVC の操作と同等です。



2.2.2. アプリケーションウィザード

新規アプリケーションの作成、および、既存アプリケーションプログラムのソースコードの使用、どちらの場合でも、アプリケーションウィザードを使用することにより、簡単にビルド環境を構築し、プロジェクトを生成することができます。

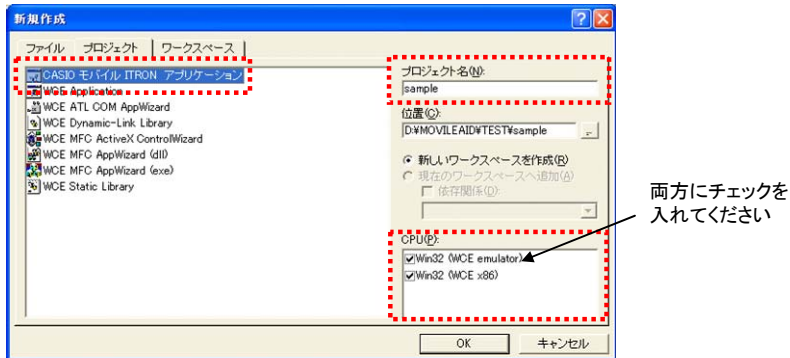
このウィザードを使用した場合、DT-930 の開発に最も適した環境を自動的に生成することが可能です。また、より詳細に細かな設定を行う場合は、ウィザード終了後に詳細設定※を行ってください。

※ 「プロジェクト(P)」 > 「設定(S)」を選択することにより、設定することができます。設定方法は、通常の eVC と同等です。

(1) プロジェクトの設定

「ファイル(F)」>「新規作成(N)」>「プロジェクト」を選択してください。

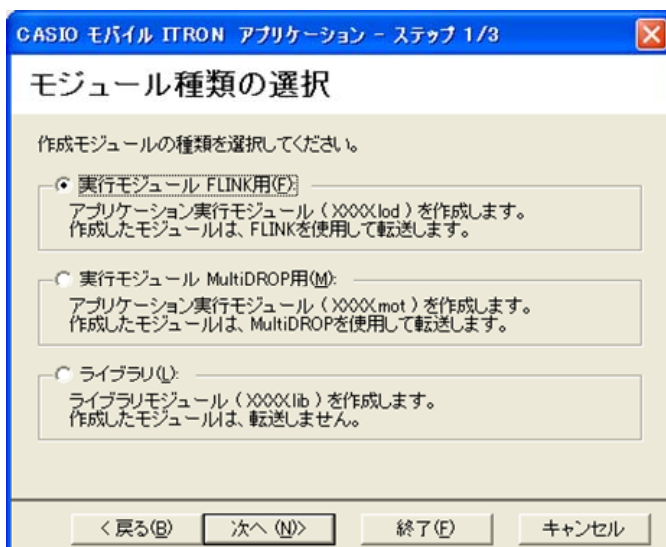
ここで、「CASIO モバイル ITRON アプリケーション」を選択し、任意のプロジェクト名を指定後、「OK」ボタンをクリックしてください。



(2) モジュール種類の選択

作成するプログラムの種類を選択してください。選択した内容に従って、自動的に実行モジュールを組み込みます。選択後「次へ(N)>」のボタンをクリックしてください。

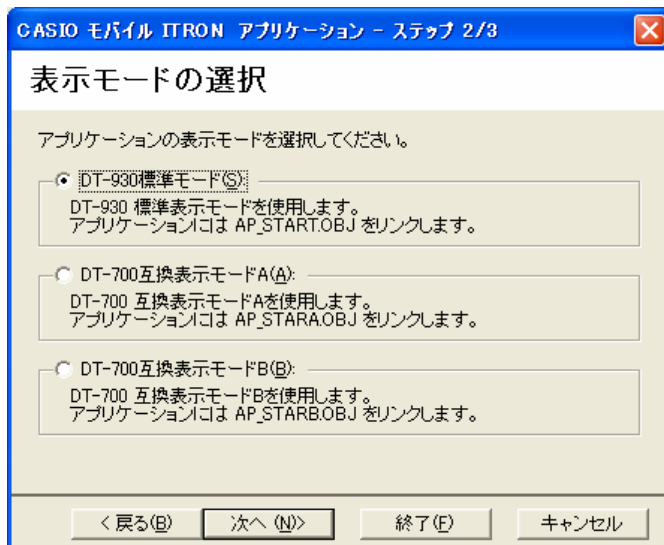
モジュール種類	説明
実行モジュール FLINK 用	FLINK で転送可能な、実行モジュール(xxxxxxx.lod)を作成します。
実行モジュール MultiDROP 用	MultiDROP で転送可能な、実行モジュール(xxxxxxx.mot)を作成します。
ライブラリ	ライブラリファイル(xxxxxxx.lib)を作成します。



(3) 表示モードの選択

使用する表示モードを選択してください。選択した内容に従って、自動的に実行モジュールを組み込みます。選択後「次へ(N)>」のボタンをクリックしてください。

表示モード	説明
DT-930 標準モード	DT-930 標準の表示モードです。 DT-900 用アプリケーションから移植する場合にも使用します。
DT-700 互換モード A	DT-700 用アプリケーションを移植する際に使用する互換表示モード A です。
DT-700 互換モード B	DT-700 用アプリケーションを移植する際に使用する互換表示モード B です。

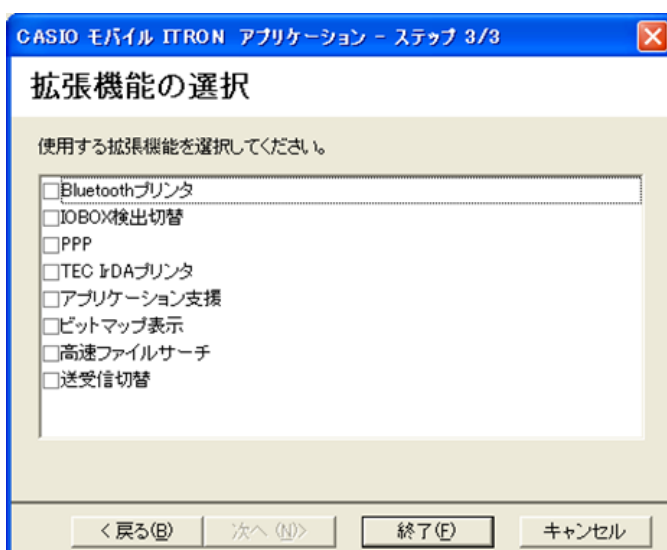


※ ライブラリを作成する場合は、この画面はスキップします。

(4) 拡張機能の選択

使用する拡張機能を、表示した一覧の中から選択してください。選択した内容に従って、自動的に実行モジュールを組み込みます。選択後「終了(F)」のボタンをクリックしてください。

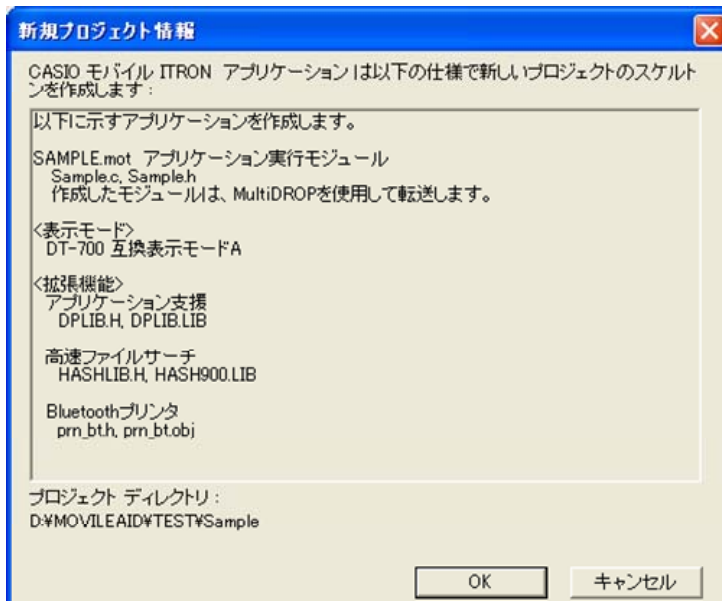
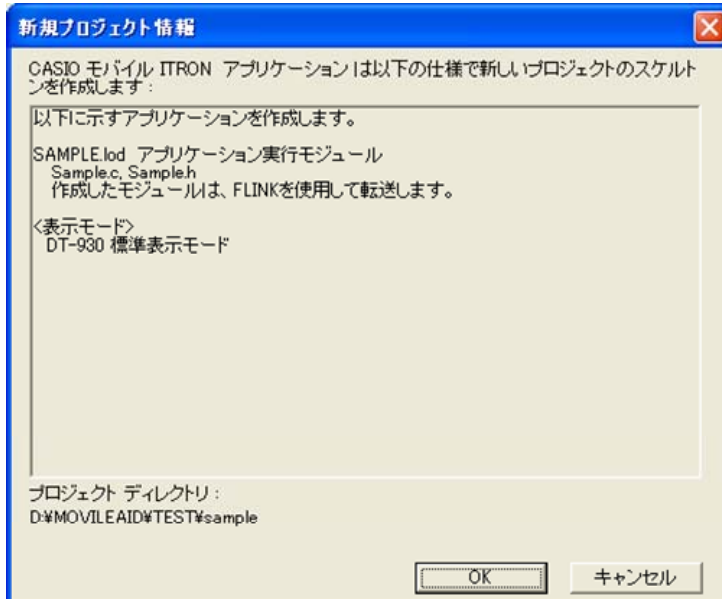
拡張機能	説明
Bluetooth プリンタ	DT-930 内蔵の Bluetooth を利用して、外部 Bluetooth ポケットプリンタを制御するための機能です。PT-200e(SATO 社製)/B-SP2D(東芝テック社製)
IOBOX 検出切替	DT-500 プロトコルによる、本機と MCU60 との IR 通信を可能にするため、IO ボックス検出機能の有効/無効を切り替える機能です。
PPP	サテライト IOBOX 経由でモデムを接続して TCP/IP 通信を実現させるための機能です。
TEC IrDA プリンタ	TEC 社製 IrDA ポケットプリンタ(DT-810PR)を制御するための機能です。
アプリケーション支援	アプリケーション支援ライブラリです。
ビットマップ表示	BMP 形式のファイルを使用してビットマップ表示を行うための機能です。
高速ファイルサーチ	HASH 関数による高速ファイルサーチ機能です。
送受信切替	赤外線通信時のデータ送受信方向の自動切換え機能です。



※ 上記の一覧表示は、初期状態で、本 SDK に組み込んでいます。独自に拡張機能を追加/削除した場合は、追加/削除した内容を表示します。

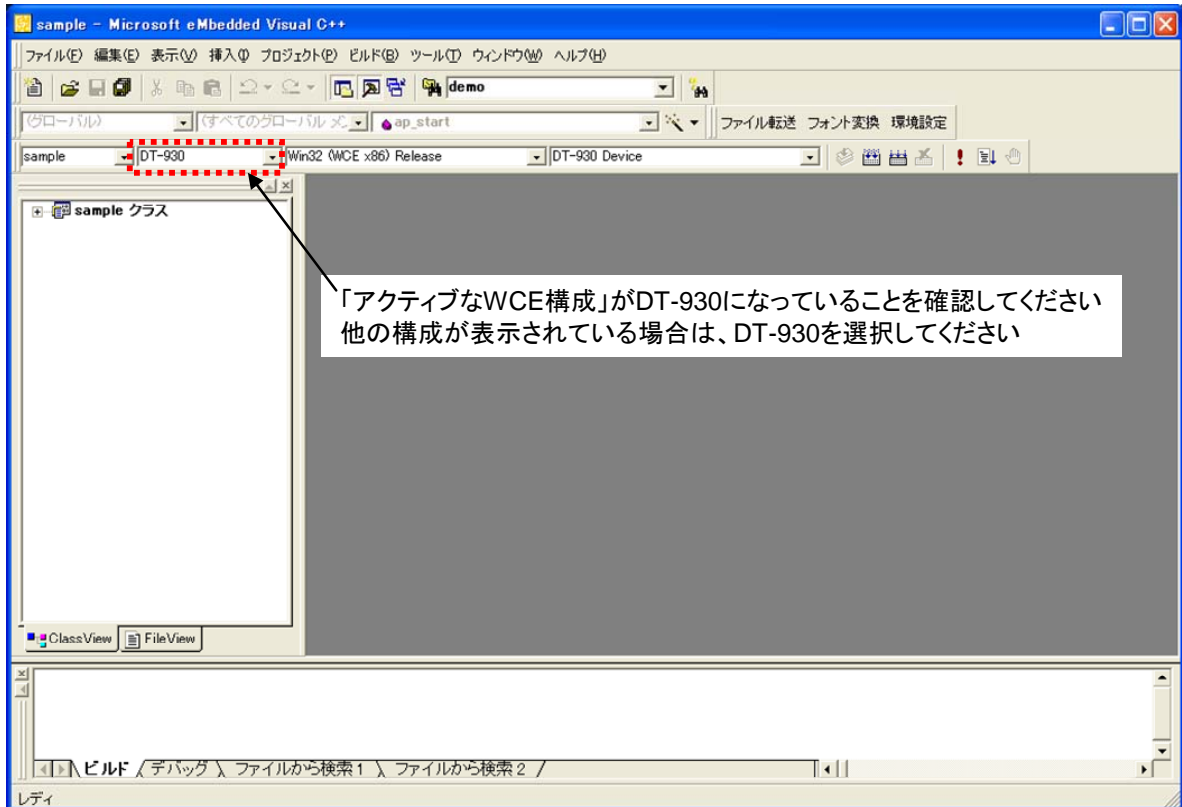
(5) 新規プロジェクト情報

ウィザードで設定された、新規プロジェクト情報を表示します。確認後、「OK」ボタンをクリックしてください。



(6) プロジェクトの完成

アプリケーションウィザードが終了すると、以下の画面を表示します。
これで、アプリケーションをビルドする環境が整いました。

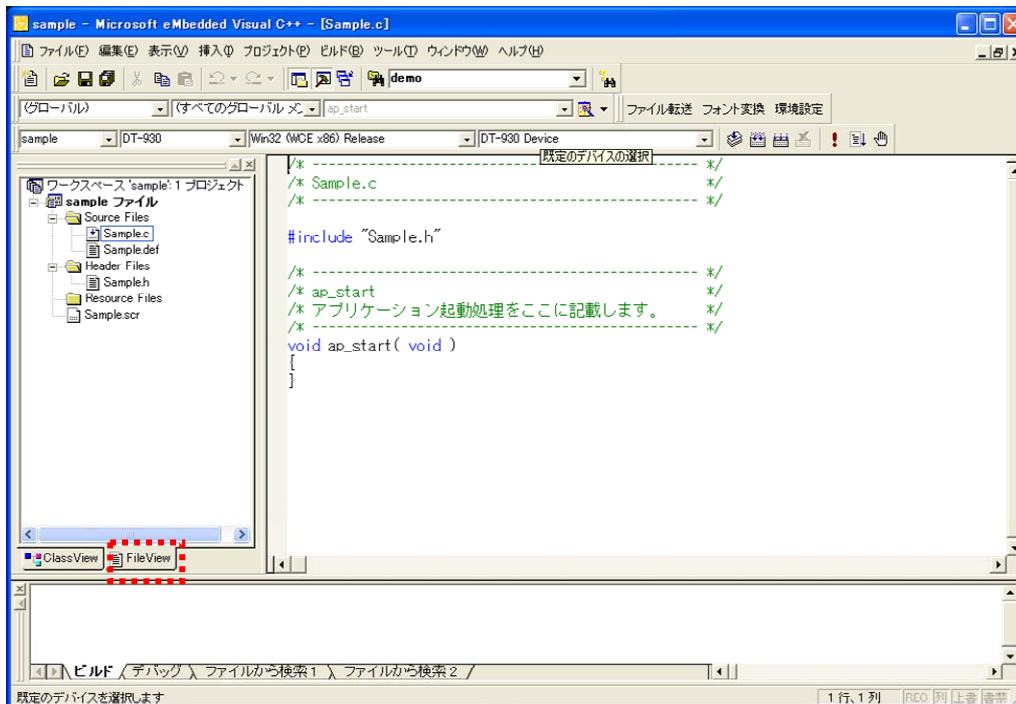


2.2.3. アプリケーションの作成/編集(新規アプリケーション)

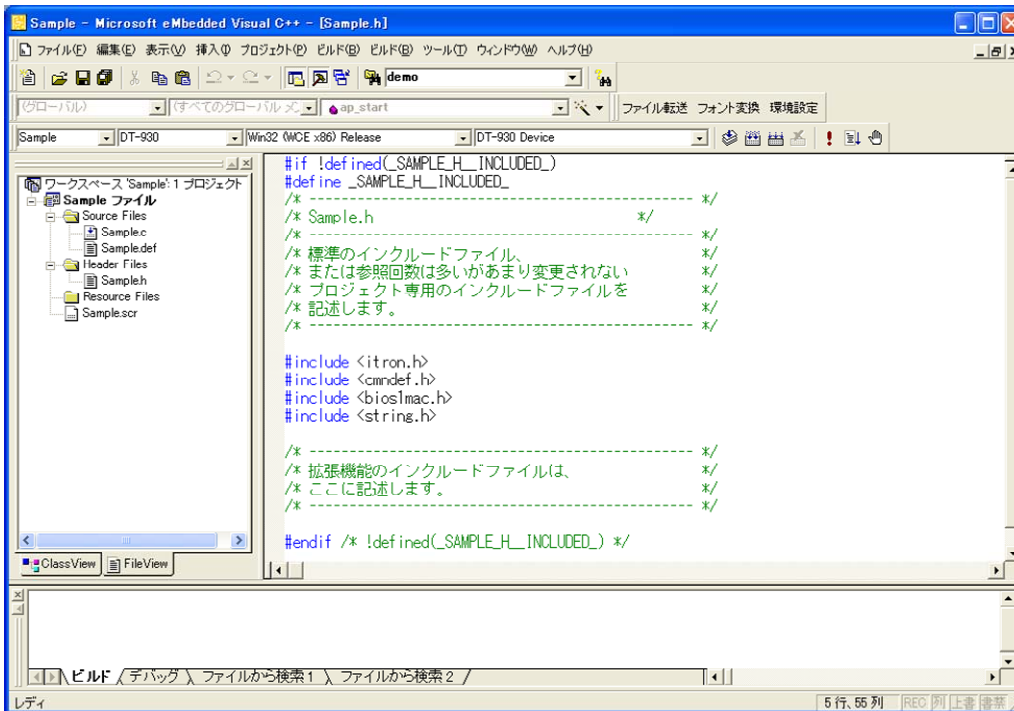
アプリケーションウィザードが完了すると、選択した構成に従って、自動的にモジュールを組み込み、ベースとなるアプリケーションソースを生成します。

「FileView」を開くと、「Source Files」「Header Files」フォルダ内に、ベースとなるソースファイルとヘッダファイルがありますので、このファイルを編集して、新規アプリケーションを作成してください。

(1) ソースファイル



(2) ヘッダファイル



```
#if !defined(_SAMPLE_H_INCLUDED_)
#define _SAMPLE_H_INCLUDED_
/* ----- */
/* Sample.h */
/* ----- */
/* 標準のインクルードファイル、 */
/* または参照回数(多し)があまり変更されない */
/* プロジェクト専用のインクルードファイルを */
/* 記述します。 */
/* ----- */

#include <itron.h>
#include <cmdf.h>
#include <bioslmac.h>
#include <string.h>

/* ----- */
/* 拡張機能のインクルードファイルは、 */
/* ここに記述します。 */
/* ----- */

#endif /* !defined(_SAMPLE_H_INCLUDED_) */
```

2.2.4. アプリケーションの作成/編集(新規/既存共通)

以下の内容は、新規のアプリケーションを作成する場合でも既存のアプリケーションを作成する場合でも共通の機能です。

(1) 複数ソースの登録

プロジェクト内に複数のソースコードを登録することが可能です。

「**Source Files**」フォルダ上で右クリックし、「**ファイルをフォルダへ追加(F)**」を選択し、登録します。

登録した場合は、自動的にビルド環境を再構成しますので、特に環境の設定をし直す必要はありません。

(2) コメントの記述方法

コメントを記載する場合に、「//」を使用することが可能です。

(3) ロードモジュール名

ビルドにより作成するロードモジュール(DT-930 で動作するモジュール)名は、プロジェクト名と同じものです。

(4) 基本操作

アプリケーションの作成/編集に要する基本操作機能は、eVC と同等です。

※ 「**Source Files**」フォルダ内にある、xxx.def (xxx はプロジェクト名)ファイルは削除しないでください。このファイルはデバッグ時にエミュレータを起動するために使用するので、削除すると、エミュレータが起動しなくなります。

2.2.5. アプリケーションの作成/編集(既存アプリケーション)

アプリケーションウィザードが完了すると、選択した構成に従って、自動的にモジュールを組み込み、ベースとなるアプリケーションソースを生成します。

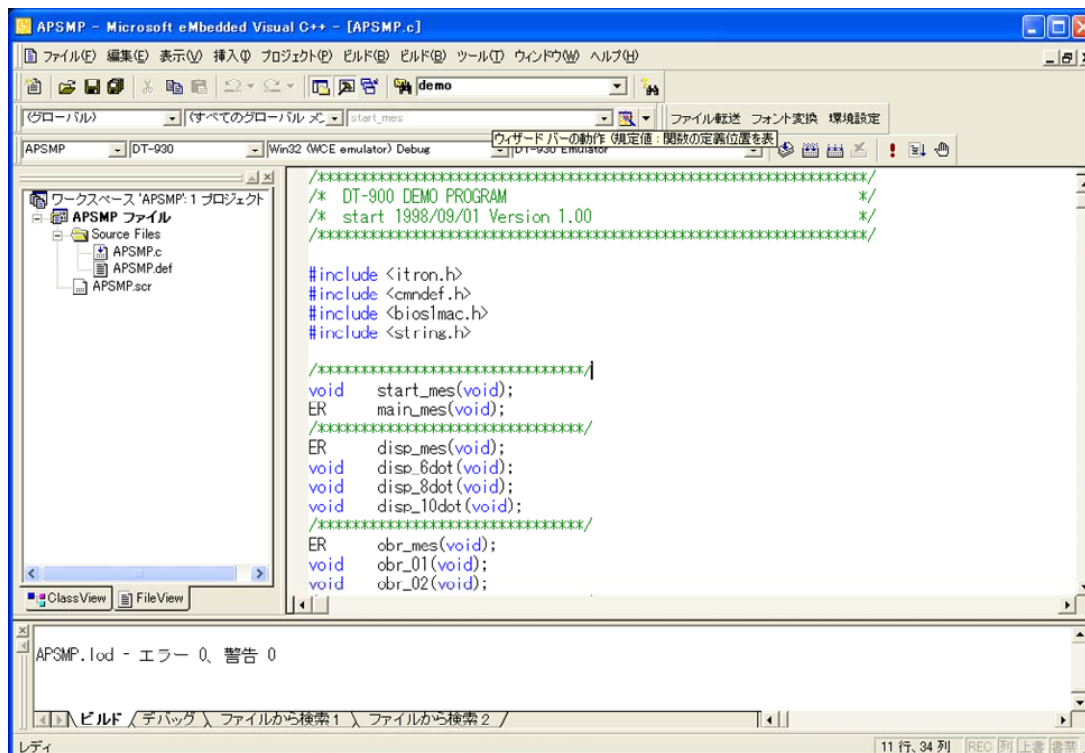
このソースをベースに、既存アプリケーション(MS/DOS版環境)を作成する方法は以下の2通りあります。

(1) ソースコードの貼り付け

「Source Files」「Header Files」フォルダ内に、ベースとなるソースファイルとヘッダファイルがありますので、既存アプリケーションのソースコードを、これらのファイルの所定の場所に追加します。

(2) ソースコードの登録

ウィザードにより生成された、ソースおよびヘッダファイルを削除し、既存アプリケーションのソースおよびヘッダファイルを「2.2.4 アプリケーションの作成/編集(新規/既存共通)」で示した手順で、登録してください。



2.3. アプリケーションのビルド

2.3.1. ビルド方法

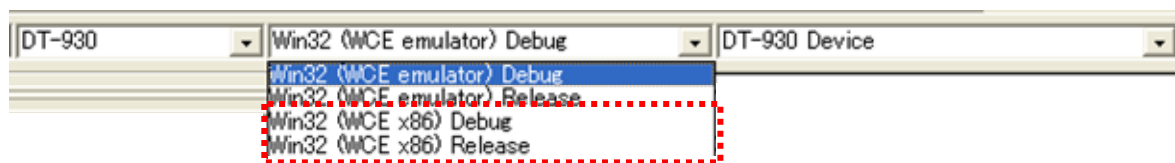
ビルド方法は、eVC と同等です。

「ビルド(B)」>「ビルド(B)」を選択します。

ただし、DT-930 で動作するロードモジュールを作成する場合と、エミュレータで動作させる場合は、「アクティブな構成」を切り替えてください。

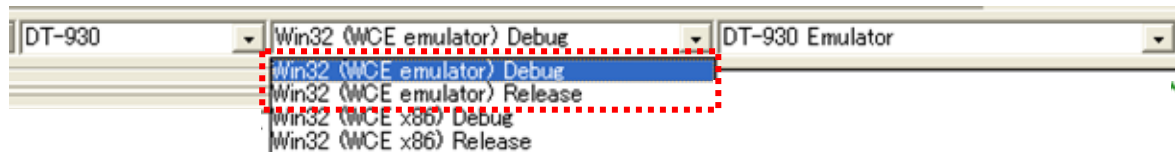
DT-930 用ロードモジュールのビルド

Win32 (WCE x86) Release または Win32 (WCE x86) Debug を選択してください。

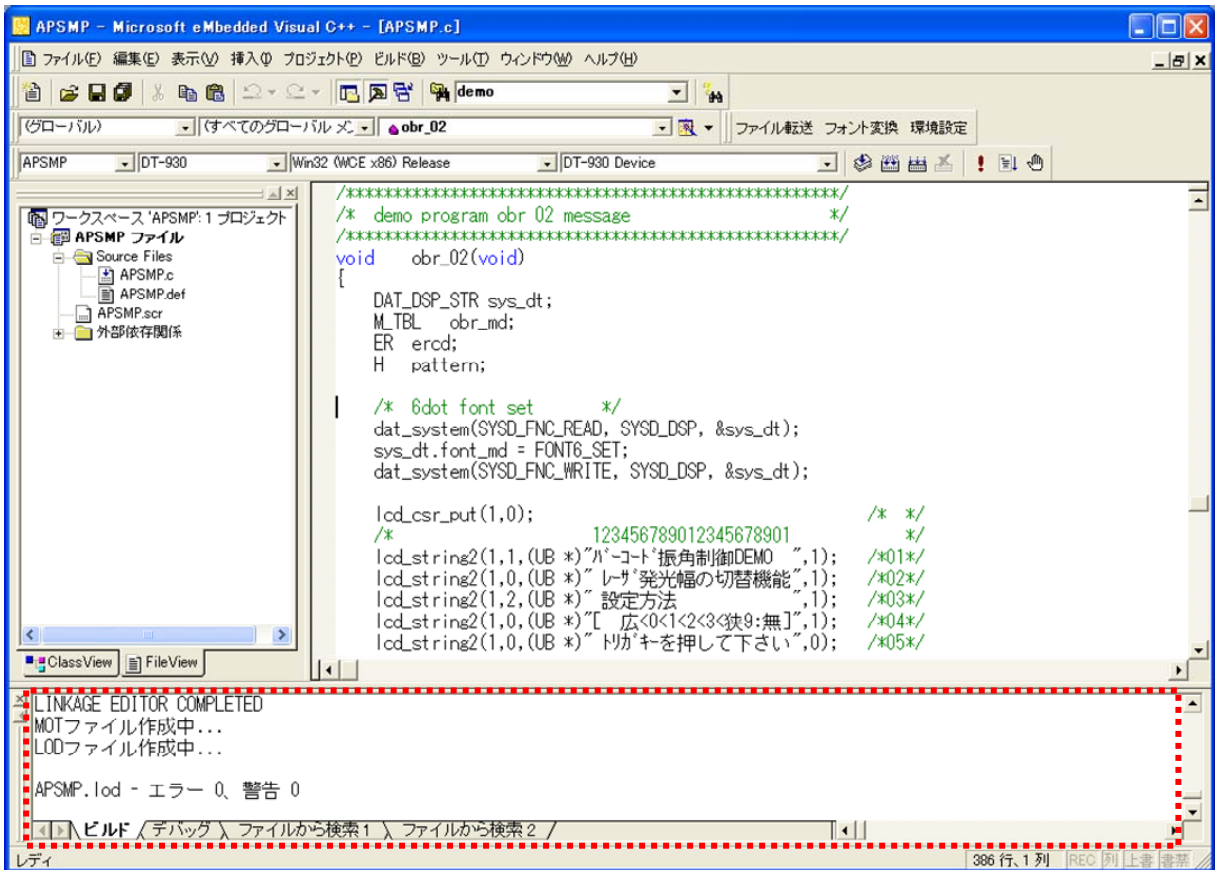


エミュレータ用ロードモジュールのビルド

Win32 (WCE emulator) Release または Win32 (WCE emulator) Debug を選択してください。



ビルドを開始すると、デバッグウィンドウにコンパイラリンクの実行状態と結果を表示します。



2.3.2. 生成するオブジェクトの種類

アプリケーションウィザードで指定した条件に応じて、規定のフォルダに実行モジュールを生成します。

モジュール種類	生成されるモジュール
実行モジュール FLINK 用	xxxxxxxx.LOD / xxxxxxxx.MAP
実行モジュール MultiDROP 用	xxxxxxxx.MOT / xxxxxxxx.MAP
ライブラリ	xxxxxxxx.LIB

2.4. アプリケーションのデバッグ

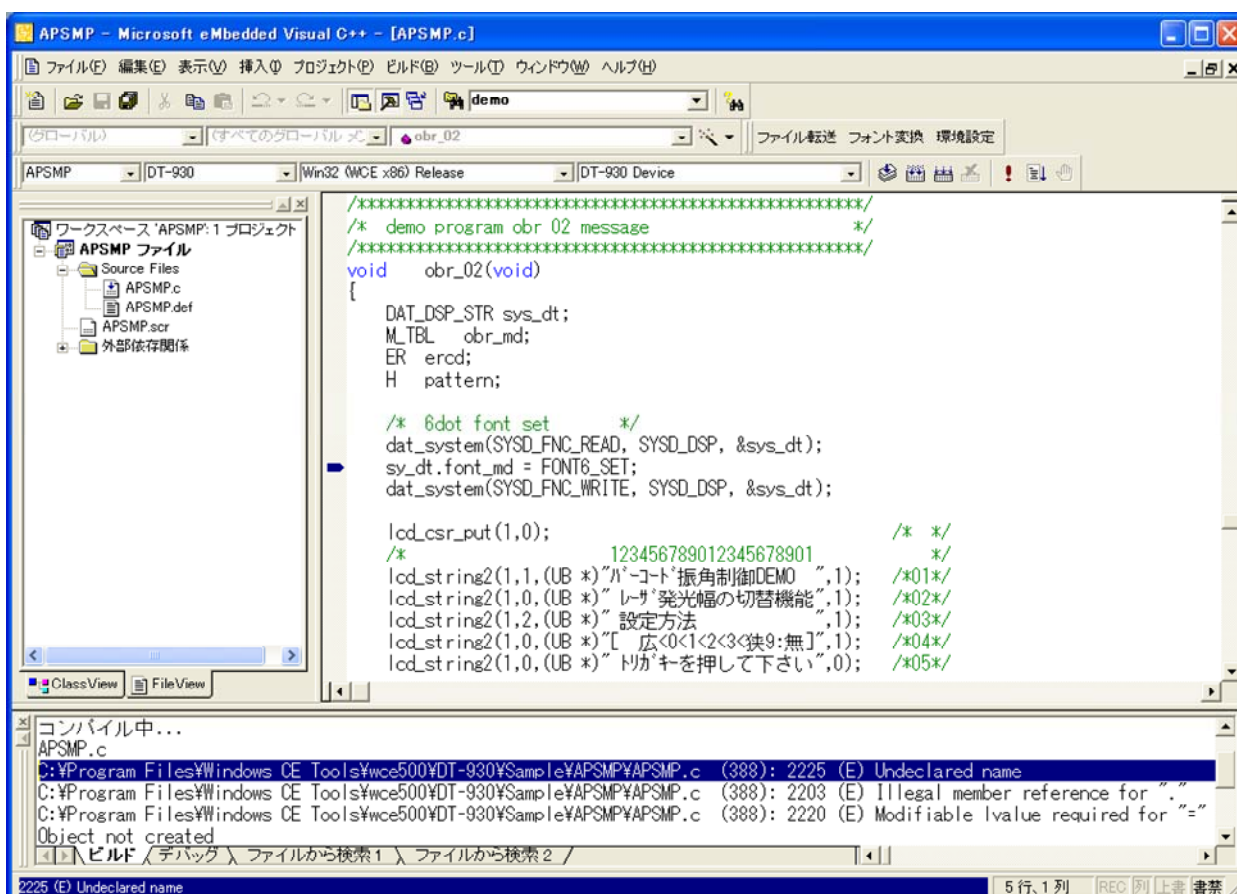
2.4.1. デバッグ基本操作

デバッグ方法は、eVC と同等です。

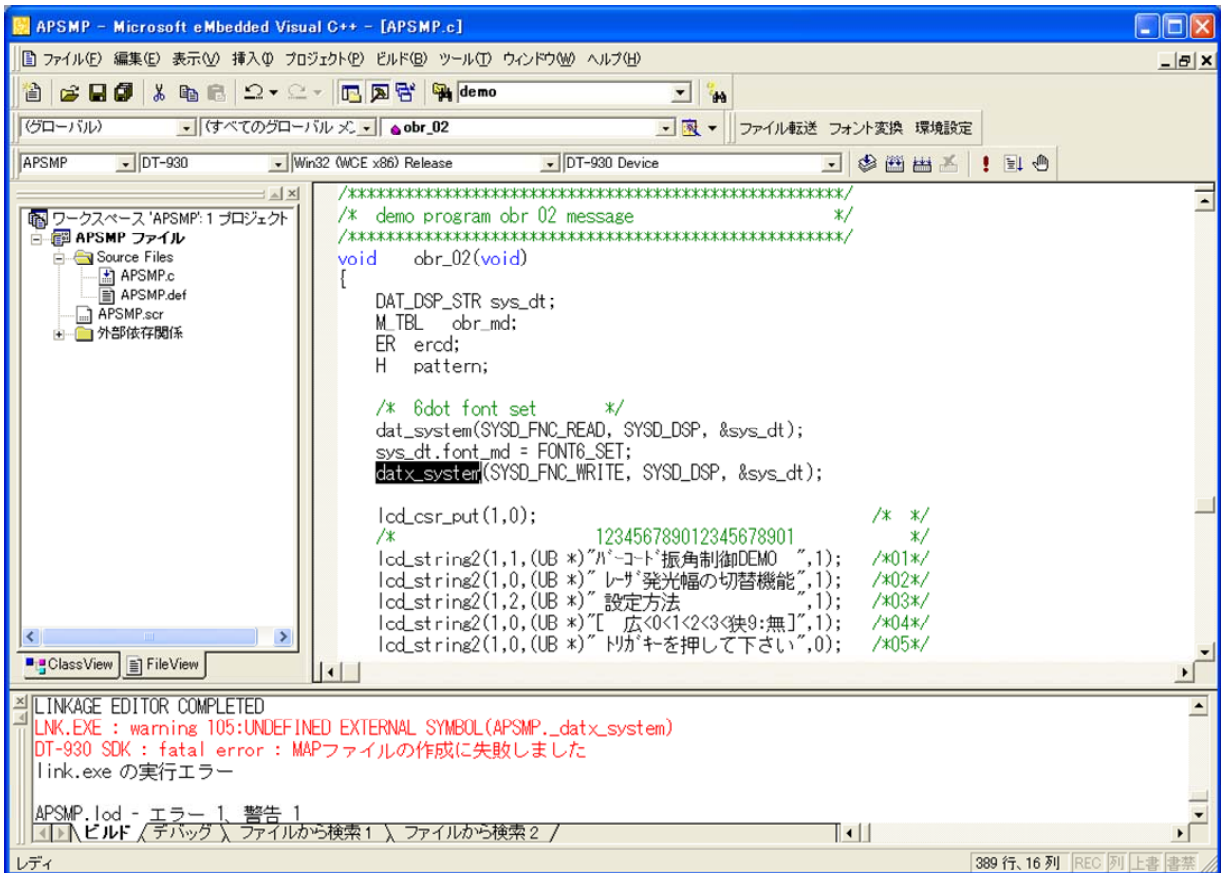
コンパイルおよびリンク時に発生したエラー情報は、すべてデバッグウィンドウに表示しますので、その内容を確認しながらデバッグを行ってください。

(1) コンパイルエラー

コンパイルエラーのメッセージにマウスをあわせてクリックすると、エラーが発生した箇所のソースコードにジャンプします。



(2) リンクエラー

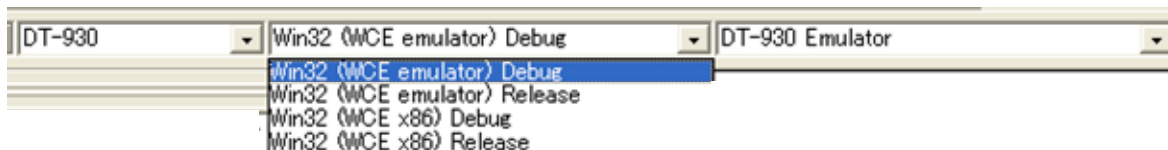


2.4.2. エミュレータデバッグ

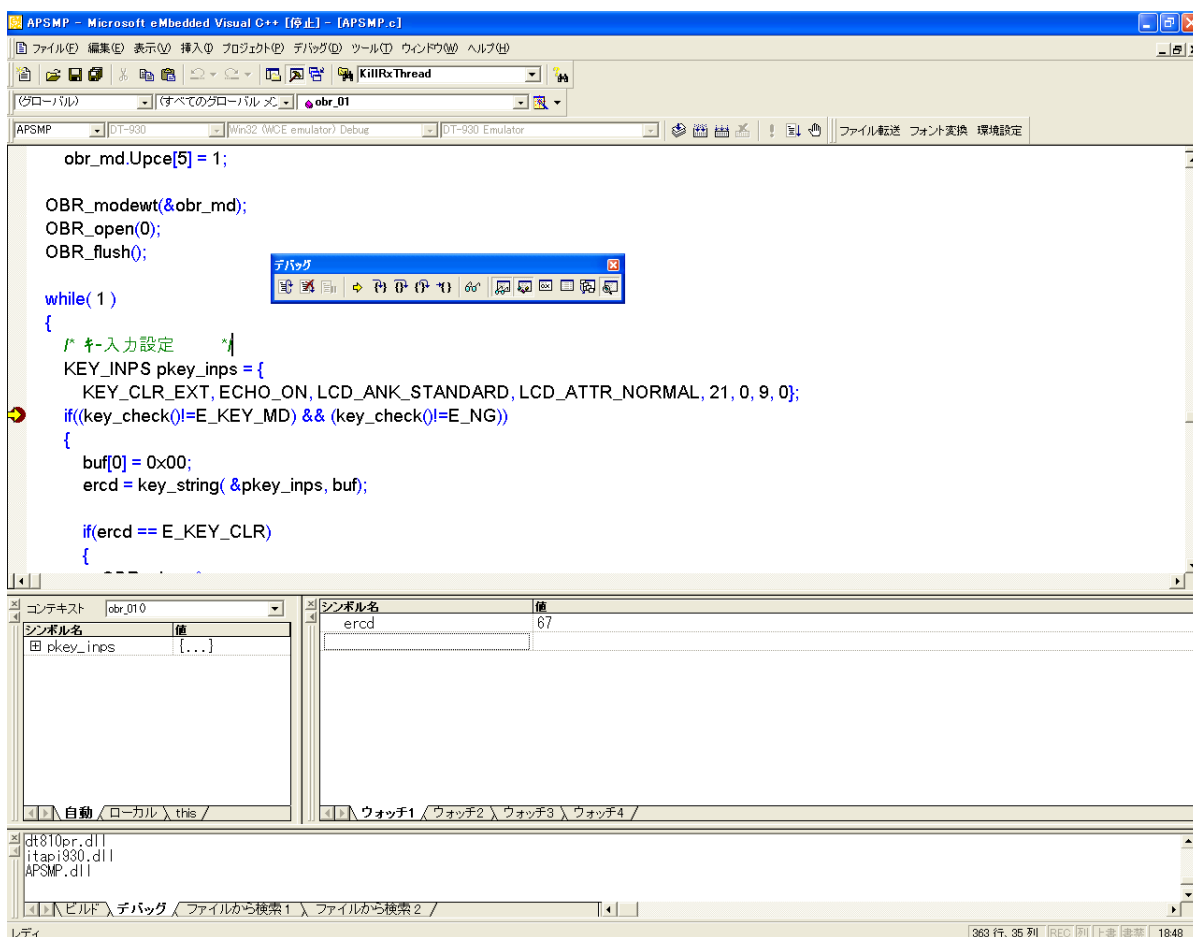
本 SDK には、DT-930 エミュレータを搭載しています。これにより、DT-930 実機での確認を行う前に、PC 上で基本的な機能のデバッグが可能となります。

(1) エミュレータ動作の切り替え

「アクティブな構成」を Win32 (WCE emulator) Debug または Win32 (WCE emulator) Release に切り替えます。



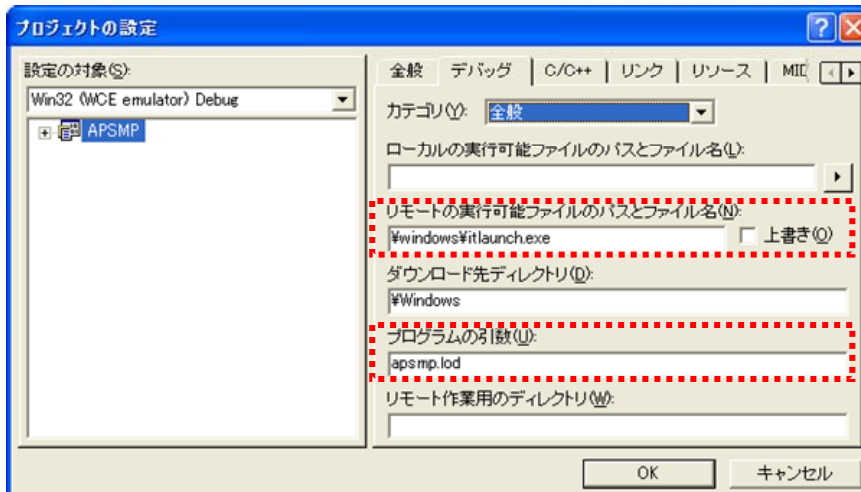
Debug を選択した場合は、ソースコードに対して、ブレークポイントを設定し、ステップ実行等によるデバッグが可能になります。



(2) プロジェクトの設定

「プロジェクト(P)」>「設定(S)」>「デバッグ」を選択し、下記の内容を設定します。

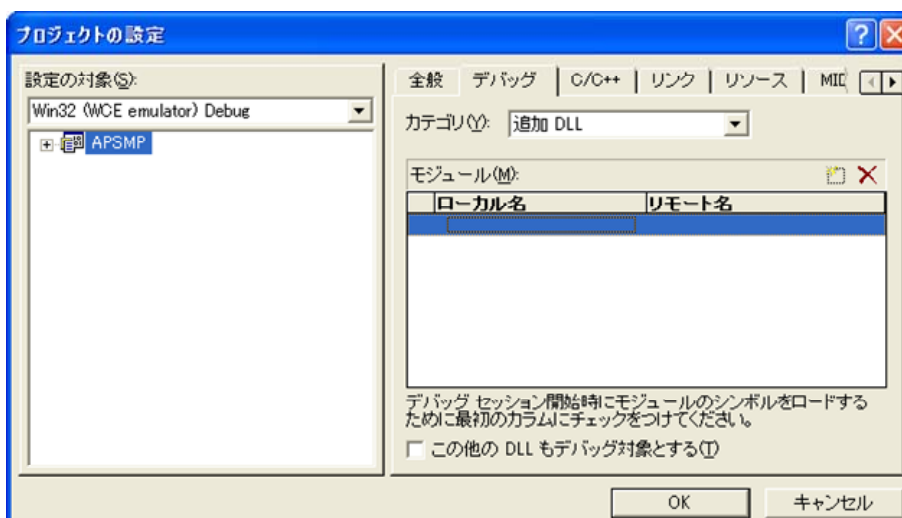
リモートの実行可能ファイルのパスとファイル名 : %windows%itleaunch.exe
プログラムの引数 : 実行モジュール名 (例: apsmplod)



(3) 内部関数のデバッグ抑止

デバッグに必要な、内部モジュールのデバッグを対象から外します。カテゴリの追加 DLL を選択すると、下図の画面を表示しますので、「この他の DLL もデバッグ対象とする」のチェックを外してください。

※ この設定をしていないと、エミュレータが起動した時点で、いくつかのローカルモジュールの検索ダイアログを表示することがあります。その場合は、キャンセルボタンをクリックしてください。



(4) デバッグ開始

「ビルド(B)」>「デバッグの開始(D)」>「実行(G)」を選択することにより、デバッグを開始します。デバッグを開始すると、エミュレータと I/O シミュレータを自動的に起動し、下図の画面を表示します。

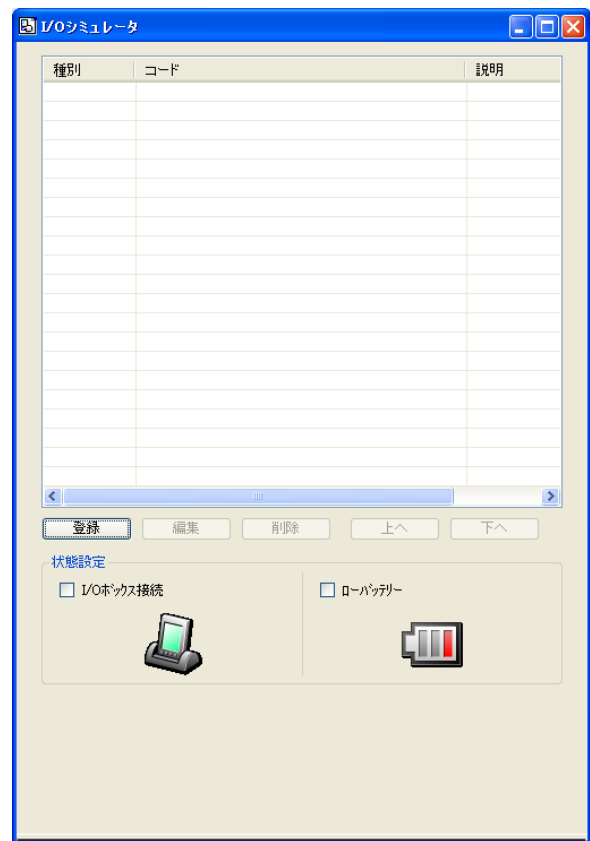
DT-930 エミュレータ

マウスや PC のキーボードによるキー入力や、実行画面の表示等、DT-930 実機上の動作を再現します。



I/O シミュレータ

バーコード入力・ローバッテリー発生・IOBOX 装着の動作を擬似的に行うものです。



詳細な機能に関しては、「3. エミュレータの使用方法」を参照してください。

2.5. アプリケーションのインストール

DT-930 実機へ、作成したアプリケーションをインストールします。

ツールバーの「ファイル転送」をクリックすると、現在のプロジェクトで設定しているロードモジュールの形式に応じて、LMWIN または、MultiDrop が起動します。

ロードモジュール	起動するファイル転送ユーティリティ
実行モジュール FLINK 用	LMWIN
実行モジュール MultiDrop 用	MultiDrop

ファイル転送ユーティリティが起動したら、規定のフォルダに生成しているロードモジュールを設定して、DT-930 へのインストールを行ってください。

※ 各ファイル転送ユーティリティの使用方法に関しては、「**4. その他の機能**」もしくは各々のマニュアルを参照してください。

3. エミュレータの使用法

3.1. DT-930 エミュレータ

マウスや PC のキーボードによるキー入力や、実行画面の表示等、DT-930 実機上の動作を再現します。



3.1.1. キー入力機能

DT-930 実機と同等のキー入力機能を搭載しています。
画面上のキーをマウスでクリックすることにより入力が可能です。
※ PC のキーボードからの入力も可能です。

3.1.2. 表示機能

DT-930 実機と同等のフォントイメージを搭載し、実機同等の表示イメージを再現します。

3.1.3. バーコード入力機能

I/O シミュレータに登録されたバーコードデータを、トリガキーの押下等、バーコードの入力操作を行うことにより、指定したバーコードデータを入力することが可能です。

※ DT-930 エミュレータのトリガキーを押下(マウスクリック)する場合は、必ず1秒以上押し続けてください。
押下(マウスクリック)する時間が短いと、正しく入力できない場合があります。

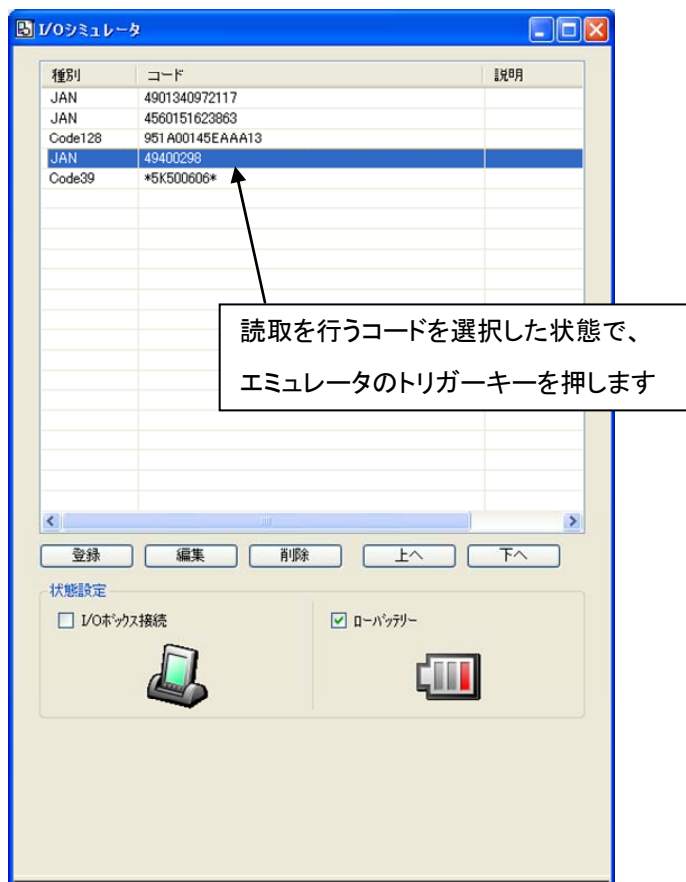
3.1.4. サウンド機能

DT-930 実機と同等の BEEP/SOUND 機能を搭載しています。

3.1.5. インジケーション機能

下記のインジケーション機能を再現します。

インジケーション	説明
LED 点灯	バーコード入力が行われた際に LED が点灯します
バイブレータ振動	DT-930 のイメージが振動します
ローバッテリーICON	ローバッテリーICON が点灯します
トリガキー押下	左右トリガキーが押されたことをより見やすくするために、押した時にトリガキーの横にマークを表示します



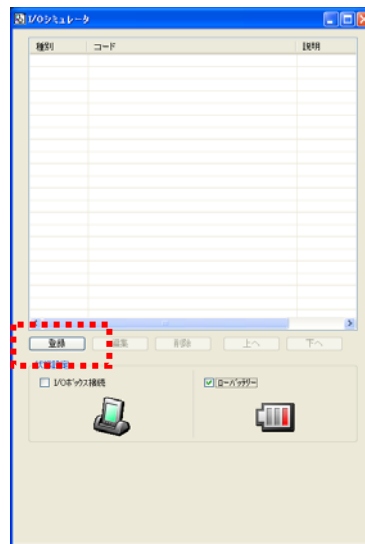
3.2. I/O シミュレータ

入力バーコードの登録・ローバッテリー発生・IOBOX 装着動作を擬似的に行うものです。

3.2.1. 入力バーコードの登録

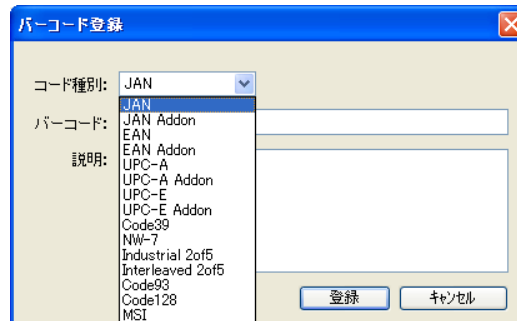
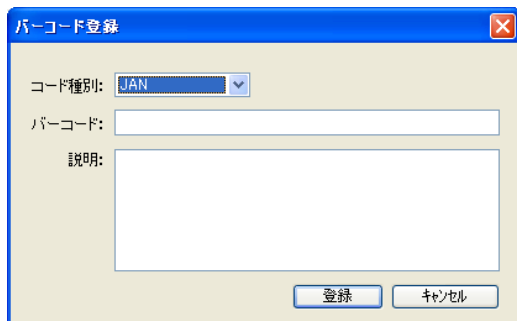
(1) 登録選択

I/O シミュレータの「登録」ボタンをクリックすると、バーコード登録画面に移ります。

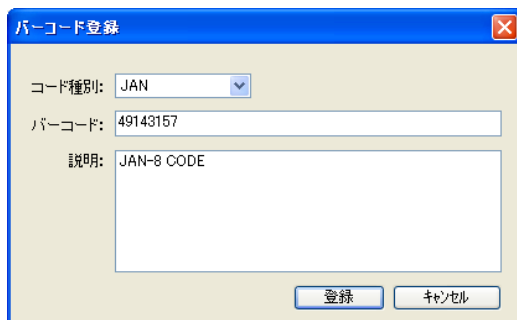


(2) バーコード登録

「コード種別」を選択し、任意のバーコード種別を選択します

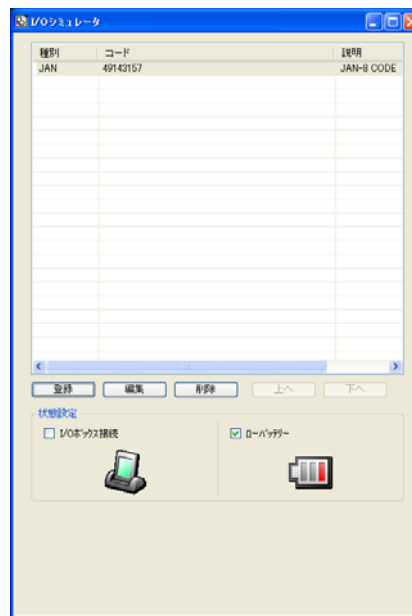


(3) バーコードのデータと説明を登録



(4) 登録完了

登録が完了すると、登録したバーコードが一覧表に表示します。この登録操作を繰り返して、デバッグに必要なバーコードを予め登録しておいてください。

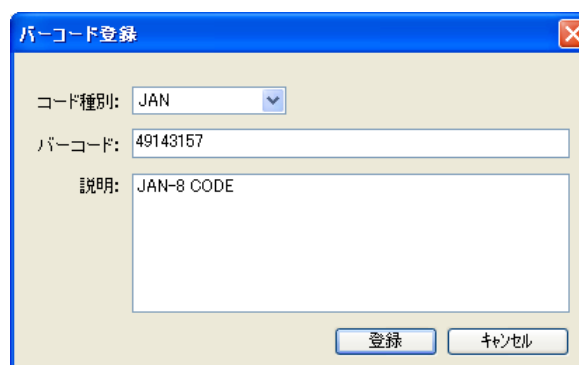


※ バーコード登録時の注意事項

登録可能なバーコードは、DT-930 エミュレータ上で読取可能となっているコードに限ります。エミュレータ起動直後はすべてのコードが読取可能となっていますが、アプリケーションを起動した場合は、そのアプリケーションで設定した読取可能コードのみが、登録可能となります。

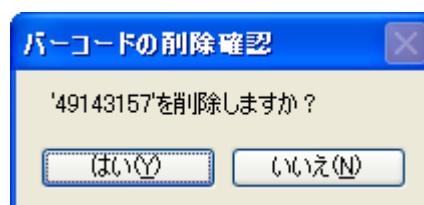
(5) 登録内容の編集

編集したいバーコードを一覧表から選択し、「編集」ボタンをクリックすると、右記の画面を表示しますので、編集してください。



(6) 登録内容の削除

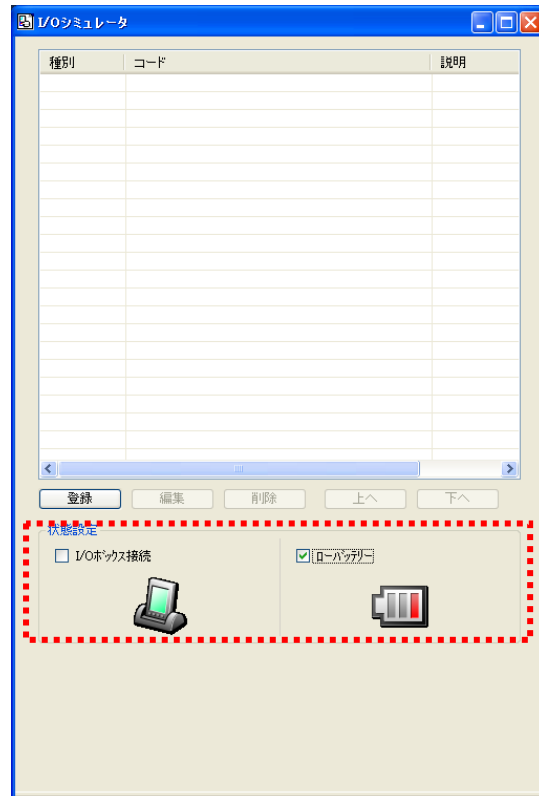
削除したいバーコードを一覧表から選択し、「削除」ボタンをクリックすると、右記の画面を表示しますので、確認後、削除してください。



3.2.2. ローバッテリー発生/IOBOX 接続動作

I/O シミュレータの状態設定にある「I/O ボックス接続」/ローバッテリー」にチェックを入れると、それぞれの動きを擬似的に再現します。

- I/O ボックス接続
DT-930 エミュレータに対して、I/O ボックスが接続されたことを通知します。
アプリケーション側で I/O ボックス接続のイベントを確認することができます。
- ローバッテリー
DT-930 エミュレータに対して、ローバッテリーが発生したことを通知します。
DT-930 エミュレータのローバッテリー ICON が点灯します。また、アプリケーション側で、ローバッテリー発生イベントを確認することができます。



3.3. 実機とエミュレータの動作の違い

DT-930 実機と、DT-930 エミュレータの動作の違いを説明します。

3.3.1. デバッグ

デバッグできるのは、エミュレータのみになります。
実機を用いてのリモートデバッグはできませんのでご注意ください。

3.3.2. ドライブレター

エミュレータ上でファイルの操作(コピーや削除等)を行う場合は、eVC に付属しているリモートファイルビューワーを使用してください。その際、ドライブレターは DT-930 実機と下記のように異なります。

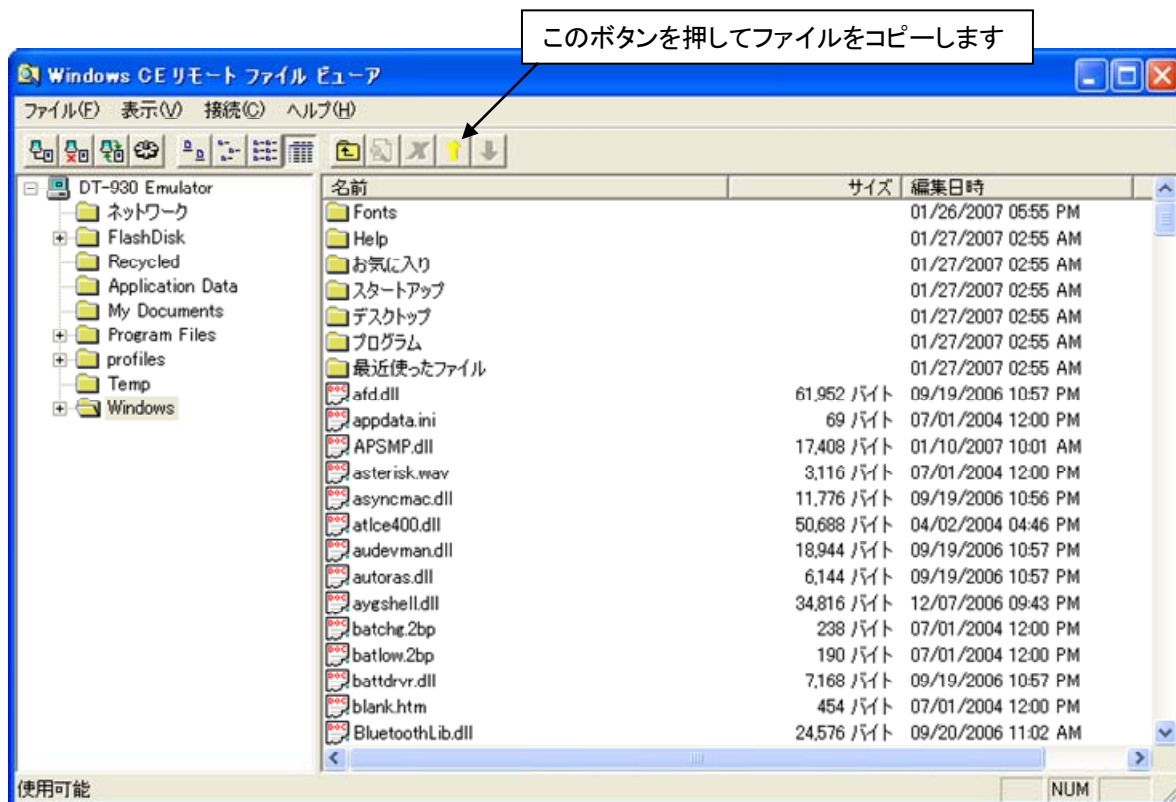
(1) FAT ファイルモードで使用する関数を実行した場合

DT-930 実機	DT-930 エミュレータ
A ドライブ	¥windows
B ドライブ	¥Flash Disk

※ プログラム内では、A:¥data.dat と記載すると、自動的に¥windows¥data.dat と読み替えを行います。

(2) DT-700 互換ファイルモードで使用する関数を実行した場合

DT-930 実機	DT-930 エミュレータ
ドライブレター無し	¥windows



3.3.3. 通信機能

赤外線および Bluetooth 等の通信機能を使用する関数は、エミュレータでは動作しません。該当する関数を実行した場合は、何もせずにエラーコードを返し、次の行に処理が移ります。

3.3.4. システムファイル

下記のシステムファイルは動作しません。

CONFIG.HTS / CONFIG.OBR / CONFIG.ID / CONFIG.PAS / ASTART.HTS / PatchXXX.LOD

3.3.5. 副電池

副電池のローバッテリー検出はできません。

3.3.6. バックライト/コントラスト

バックライトおよびコントラストは変化しません。

3.3.7. DT-700 互換表示モード

DT-700 互換表示モードの登録、および、ビルドはできますが、機能しません。(DT-930 の表示モードで動作します)

3.3.8. ユーザフォント/外字

DT-930 エミュレータでユーザフォントおよび外字フォントを使用する場合は、お手持ちのフォントをあらかじめ TrueType フォントに変換する必要があります。

TrueType フォントへの変換については、「**4.1 フォントコンバータ**」を参照してください。

3.3.9. 関数インタフェース

(1) デバイス制御ライブラリ

関数	実機動作との違い
システムデータ管理	
dat_system	電源: 値の設定/読出のみ 日本語/英語モード: 値の設定/読出のみ コントラスト設定: 値の設定/読出のみ 通信: 値の設定/読出のみ 機器 ID: CONFIG.ID の値を返却/無い場合は「??????」を返却 BIOS バージョン: 固定値 1.0000 を返却 パッチバージョン: 固定値 1.0000 を返却
dat_OSVer_Read	DT930 エミュレータのバージョン/日付を返却
dat_dealer_chk	Itlaunch.exe の起動パラメータに設定した値が有効 設定しない場合は E_OK を返却
dat_mem_size	-
電源管理	
pwr_hold_apo	-
pwr_off	レジューム設定は無効
pwr_ioboxBootMode	E_NG を返却(動作無し)
ブザー鳴動	
s_beep	-
s_sound	-
バイブレータ	
pwr_vibrator	-
日時設定	
s_dateset	-
s_dateget	-
s_timeset	-
s_timeget	-
ファイル管理(ドライブレターが異なります)	
fil_mkdir	-
fil_rmdir	-
fil_remove	-
fil_rename	-
fil_fstat	-
fil_chsize	-
fil_getsize	-
fil_findfirst	-
fil_findnext	-
fil_filesize	-
fil_filefind	-
dat_fdir	-
dat_fdel	-
dat_frname	-
dat_F_Search	-

関数	実機動作との違い
open	-
close	-
read	-
write	-
lseek	-
sbrk	-
イベント通知機能	
pwr_inhabit	LB1 の通知のみ
pwr_inhabit_clr	-
key_fnc_mode	-
s_settimer	-
s_timerend	-
s_settimer2	-
s_timerend2	-
flg_sts	-
clr_flg	-
wai_flg	タイマーフラグ待ちのみ
画面表示	
lcd_cls	-
lcd_csr_set	-
lcd_csr_put	-
lcd_csr_get	-
lcd_char	ESC(画面クリア/カーソル位置設定)非対応
lcd_string	ESC(画面クリア/カーソル位置設定)非対応
lcd_string2	ESC(画面クリア/カーソル位置設定)非対応
lcd_userstr	ESC(画面クリア/カーソル位置設定)非対応
lcd_line	-
lcd_gaiji	切替により正常終了しますが、指定したフォント表示はできません
lcd_usrfont	切替により正常終了しますが、指定したフォント表示はできません
lcd_romfont	-
lcd_led	-
lcd_el	バックライトの明るさは変化しません
キー制御	
key_select	-
key_read	-
key_string	-
key_num	-
key_check	-
key_clear	-
key_fnc	-
OBR 制御	
OBR_open	-
OBR_close	-
OBR_getc	-

関数	実機動作との違い
OBR_gets	-
OBR_flush	-
OBR_stat	-
OBR_moderd	-
OBR_modewt	-
OBR_chgbuf	-
OBR_trigmode	E_NG を返却(動作無し)
OBR_swing	値の設定/読出のみ
OBR_widenarrow	値の設定/読出のみ
OBR_getadjust	値の読出のみ
OBR_setadjust	値の設定のみ
OBR_getmargincheck	値の読出のみ
OBR_setmargincheck	値の設定のみ
シリアル通信制御	
c_open	E_NG を返却(動作無し)
c_close	E_NG を返却(動作無し)
c_status	E_NG を返却(動作無し)
c_hold	E_NG を返却(動作無し)
c_chkopen	E_NG を返却(動作無し)
c_dout	E_NG を返却(動作無し)
c_din	E_NG を返却(動作無し)
c_tmddin	E_NG を返却(動作無し)
c_out	E_NG を返却(動作無し)
c_break	E_NG を返却(動作無し)
c_txx	E_NG を返却(動作無し)
c_iobox	E_NG を返却(動作無し)
c_irout	E_NG を返却(動作無し)
c_flush	E_NG を返却(動作無し)
c_bfsts	E_NG を返却(動作無し)
c_errbfring	E_NG を返却(動作無し)
c_rdersts	E_NG を返却(動作無し)
c_chghdr	E_NG を返却(動作無し)
c_timer	E_NG を返却(動作無し)
c_er	E_NG を返却(動作無し)
c_rs	E_NG を返却(動作無し)
c_errs	E_NG を返却(動作無し)
c_brkevent	E_NG を返却(動作無し)
IrDA 制御	
Ir_Open	E_IRNG を返却(動作無し)
Ir_Close	E_IRNG を返却(動作無し)
Ir_Read	E_IRNG を返却(動作無し)
Ir_Write	E_IRNG を返却(動作無し)
Ir_QueryTx	E_IRNG を返却(動作無し)
Ir_QueryRx	E_IRNG を返却(動作無し)

関数	実機動作との違い
Ir_EROn	E_IRNG を返却(動作無し)
Ir_EROff	E_IRNG を返却(動作無し)
Ir_RSON	E_IRNG を返却(動作無し)
Ir_RSOff	E_IRNG を返却(動作無し)
Ir_BreakOn	E_IRNG を返却(動作無し)
Ir_BreakOff	E_IRNG を返却(動作無し)
Ir_CheckCD	E_IRNG を返却(動作無し)
Ir_CheckDR	E_IRNG を返却(動作無し)
Ir_CheckCS	E_IRNG を返却(動作無し)
Ir_CheckCI	E_IRNG を返却(動作無し)
Ir_CheckBreak	E_IRNG を返却(動作無し)
Ir_Err_Get	E_IRNG を返却(動作無し)
Ir_State_Set	E_IRNG を返却(動作無し)
Ir_SetPortConfig	E_IRNG を返却(動作無し)
Ir_Init	E_IRNG を返却(動作無し)
Ir_SetWinMode	E_IRNG を返却(動作無し)
Bluetooth 制御	
BT_Start	E_BTNG を返却(動作無し)
BT_Stop	E_BTNG を返却(動作無し)
BT_GetLocalInfo	E_BTNG を返却(動作無し)
BT_SetLocalInfo	E_BTNG を返却(動作無し)
BT_Inquiry	E_BTNG を返却(動作無し)
BT_GetDevInfo	E_BTNG を返却(動作無し)
BT_GetDevName	E_BTNG を返却(動作無し)
BT_SetPassKey	E_BTNG を返却(動作無し)
BT_SelectDev	E_BTNG を返却(動作無し)
BT_Open	E_BTNG を返却(動作無し)
BT_Close	E_BTNG を返却(動作無し)
BT_Read	E_BTNG を返却(動作無し)
BT_Write	E_BTNG を返却(動作無し)
BT_QueryRx	E_BTNG を返却(動作無し)
BT_SaveDevInfo	E_BTNG を返却(動作無し)
BT_LoadDevInfo	E_BTNG を返却(動作無し)
BT_Err_Get	E_BTNG を返却(動作無し)
通信ユーティリティ制御(共通ファンクション)	
cu_stopKeySet	E_NG を返却(動作無し)
cu_setDrive	E_NG を返却(動作無し)
通信ユーティリティ制御(マルチドロッププロトコル)	
cu_open	E_NG を返却(動作無し)
cu_fileSend	E_NG を返却(動作無し)
cu_fileSendSet	E_NG を返却(動作無し)
cu_fileSend1	E_NG を返却(動作無し)

関数	実機動作との違い
cu_fileRecv	E_NG を返却(動作無し)
cu_msg_Send	E_NG を返却(動作無し)
cu_end	E_NG を返却(動作無し)
cu_close	E_NG を返却(動作無し)
cu_readErrStat	E_NG を返却(動作無し)
cu_readDIRjInfo	E_NG を返却(動作無し)
通信ユーティリティ制御(FLNK プロトコル)	
cu_open	E_NG を返却(動作無し)
cu_fileSend	E_NG を返却(動作無し)
cu_fileAdd	E_NG を返却(動作無し)
cu_fileRecv	E_NG を返却(動作無し)
cu_close	E_NG を返却(動作無し)
cu_readErrStat	E_NG を返却(動作無し)
cu_idle	E_NG を返却(動作無し)
cu_cmdRecv	E_NG を返却(動作無し)
cu_fileDelete	E_NG を返却(動作無し)
cu_fileMove	E_NG を返却(動作無し)
cu_makeDir	E_NG を返却(動作無し)
cu_getFileInfo	E_NG を返却(動作無し)
cu_setFileInfo	E_NG を返却(動作無し)
cu_getDiskInfo	E_NG を返却(動作無し)
cu_dateTime	E_NG を返却(動作無し)
cu_getSysInfo	E_NG を返却(動作無し)
cu_msgSend	E_NG を返却(動作無し)
cu_beep	E_NG を返却(動作無し)
cu_setloboxInfo	E_NG を返却(動作無し)
cu_fchklog_Create	E_NG を返却(動作無し)
cu_fchklog_Check	E_NG を返却(動作無し)
通信ユーティリティ制御(DT500 プロトコル)	
cu_open	E_NG を返却(動作無し)
cu_fileSend	E_NG を返却(動作無し)
cu_fileRecv	E_NG を返却(動作無し)
cu_close	E_NG を返却(動作無し)
cu_readErrStat	E_NG を返却(動作無し)
cu_SetCode	E_NG を返却(動作無し)
共通関数	
dat_Apload	呼び出し元のアプリケーションは自動終了しません ※関数実行後 ap_start()から直ちに処理を抜けるようにしてください
abort	-
exit	-
wkup_cost	メッセージ BOX が表示されます
wkup_calib	メッセージ BOX が表示されます

(2) 拡張機能ライブラリ

関数	実機動作との違い
アプリケーション支援ライブラリ	
ht_CheckDate	-
ht_CheckYear	-
ht_ConvertYear	-
ht_Checksum	-
ht_CalcCRC_ANSI	-
ht_CalcCRC_CCITT	-
ht_CalcCRC_X	-
ht_CalcLRC	-
ht_CheckCD	-
ht_FCWait	-
ht_StrInp	-
ht_NumInp	-
ht_DateInp	-
ht_TimeInp	-
ht_ShiftMode	-
ht_MLTsend	E_NG を返却(動作無し)
ht_MLTrecv	E_NG を返却(動作無し)
ht_FLNKsend	E_NG を返却(動作無し)
ht_FLNKrecv	E_NG を返却(動作無し)
lr_c_open	E_NG を返却(動作無し)
lr_c_close	E_NG を返却(動作無し)
lr_c_status	E_NG を返却(動作無し)
lr_c_hold	E_NG を返却(動作無し)
lr_c_chkopen	E_NG を返却(動作無し)
lr_c_dout	E_NG を返却(動作無し)
lr_c_din	E_NG を返却(動作無し)
lr_c_tmdin	E_NG を返却(動作無し)
lr_c_out	E_NG を返却(動作無し)
lr_c_break	E_NG を返却(動作無し)
lr_c_trxr	E_NG を返却(動作無し)
lr_c_iobox	E_NG を返却(動作無し)
lr_c_irout	E_NG を返却(動作無し)
lr_c_timer	E_NG を返却(動作無し)
lr_c_rs	E_NG を返却(動作無し)
lr_c_er	E_NG を返却(動作無し)
lr_c_errs	E_NG を返却(動作無し)
lr_c_flush	E_NG を返却(動作無し)
lr_c_bfsts	E_NG を返却(動作無し)
lr_c_errbfiring	E_NG を返却(動作無し)
lr_c_r derrsts	E_NG を返却(動作無し)
lr_c_chghdr	E_NG を返却(動作無し)
ht_fileopen	-
ht_fileclose	-

関数	実機動作との違い
ht_fileread	-
ht_filewrite	-
ht_filesize	-
ht_filelof	-
ht_waitmsec	-
ht_dbgsendmsg	動作無し
ht_beep	-
xy_modem	エラー終了(動作無し)
lr_xy_modem	エラー終了(動作無し)
ビットマップ表示ライブラリ	
bmp_iDisplayBmpImage	-
bmp_iDisplayBmpData	-
Bluetooth プリンタライブラリ	
prn_BTinf_open	E_NG を返却(動作無し)
prn_BTinf_close	E_NG を返却(動作無し)
prn_BTinf_send	E_NG を返却(動作無し)
prn_BTinf_rcvSts	E_NG を返却(動作無し)
高速ファイルサーチライブラリ	
iHashAssign	-
iHashRead	-
iHashWrite	-
iHashAdd	-
TEC IrDA プリンタライブラリ	
prn_tecinf_open	E_NG を返却(動作無し)
prn_tecinf_close	E_NG を返却(動作無し)
prn_tecinf_send	E_NG を返却(動作無し)
prn_tecinf_send2	E_NG を返却(動作無し)
prn_tecinf_status	E_NG を返却(動作無し)
送受信切替ライブラリ	
linkmodule_c_dout	E_NG を返却(動作無し)
linkmodule_c_out	E_NG を返却(動作無し)
IOBOX 検出切替ライブラリ	
io_detect	E_NG を返却(動作無し)

4. その他の機能

4.1. フォントコンバータ

DT-930 C Language SDK GUI Version はアドインツールとして「フォントコンバータ」を搭載しています。DT-930 エミュレータと実機ではフォントファイルのフォーマットが異なるため、DT-930 エミュレータでユーザ定義フォントファイル※を使用する場合は、DT-930 実機で使用しているユーザ定義フォントファイルをDT-930 エミュレータ用のユーザ定義フォントファイルに変換する必要があります。また、本ツールを使用して、新規にユーザ定義フォントファイル(for DT-900、DT-930、DT-930 エミュレータ)を作成することが可能です。

※ ユーザ定義フォントファイル

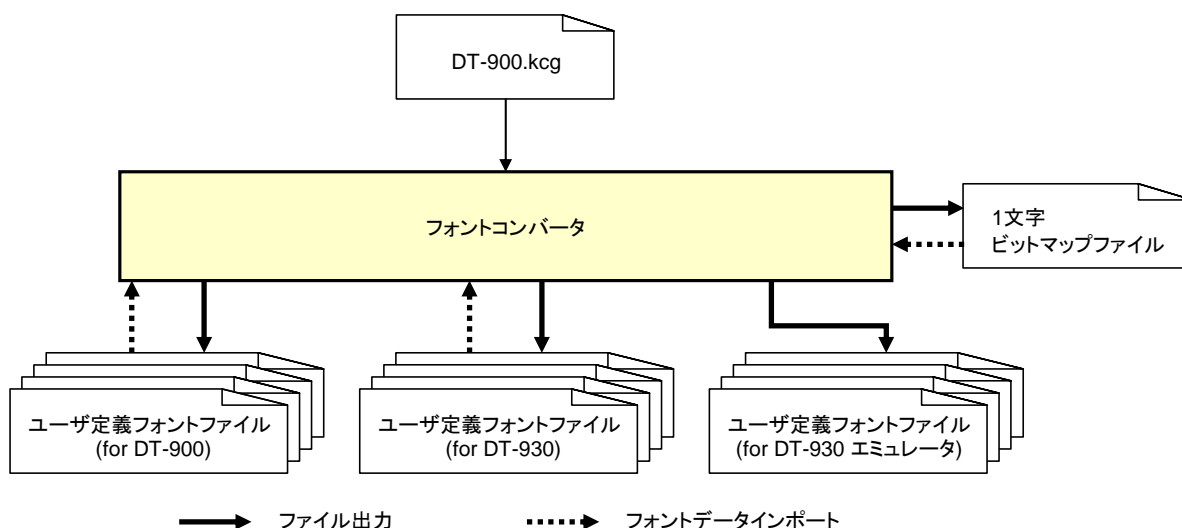
縮小 ANK(6×6)、縮小 ANK(8×8)、縮小 ANK(10×10)

標準 ANK(6×12)、標準 ANK(8×16)、標準 ANK(10×20)

漢字(12×12)、漢字(16×16)、漢字(20×20)

外字(12×12)、外字(16×16)、外字(20×20)

フォントコンバータの構成



DT-900.kcg

DT-900.kcg は DT-930 C Language SDK GUI Version と同時にインストールされます。

ユーザ定義フォントファイル(for DT-900、DT-930、DT-930 エミュレータ)を新規に作成する場合は、本ファイルをもとにフォントデータを作成します。

ユーザ定義フォントファイル(for DT-900)

ユーザ定義フォントファイル(for DT-900)は DT-900 実機で使用するユーザ定義フォントファイルです。

既にユーザ定義フォントファイル(for DT-900)がある場合は、その編集および DT-930、DT-930 エミュレータ用のユーザ定義フォントファイルに変換することが可能です。

また、DT-900.kcg を使用して新規に DT-900 用のユーザ定義フォントファイルを作成することが可能です。

ユーザ定義フォントファイル(for DT-930)

ユーザ定義フォントファイル(for DT-930)は DT-930 実機で使用するユーザ定義フォントファイルです。既にユーザ定義フォントファイル(for DT-930)がある場合は、その編集および DT-900、DT-930 エミュレータ用のユーザ定義フォントファイルに変換することが可能です。また、DT-900.kcg を使用して新規に DT-930 用のユーザ定義フォントファイルを作成することが可能です。

ユーザ定義フォントファイル(for DT-930 エミュレータ)

ユーザ定義フォントファイル(for DT-930 エミュレータ)は DT-930 エミュレータで使用するユーザ定義フォントファイルです。既にユーザ定義フォントファイル(for DT-900 または DT-930)がある場合は、それをもとに作成することが可能です。また、DT-900.kcg を使用して新規に DT-930 エミュレータ用のユーザ定義フォントファイルを作成することが可能です。

1 文字ビットマップファイル

編集したい 1 文字を選択し、その文字をビットマップファイル(*.bmp)として出力することが可能です。また、出力したビットマップファイルを外部のビットマップエディタで編集し、各ユーザ定義フォントファイルに反映することが可能です。

4.1.1. 機能

フォントデータ表示／反映

作成または編集したユーザ定義フォントファイルを確認するための Viewer 機能です。

既存のユーザ定義フォントファイル (for DT-900、DT-930) を表示する場合は、以下のファイル名および拡張子に修正してください。

- ファイル名

ファイル名	内容
ANK6	縮小 ANK (6×6)
ANK8	縮小 ANK (8×8)
ANK10	縮小 ANK (10×10)
ANK12	標準 ANK (6×12)
ANK16	標準 ANK (8×16)
ANK20	標準 ANK (10×20)
Kanji12	漢字 (12×12)
Kanji16	漢字 (16×16)
Kanji20	漢字 (20×20)
Gaiji12	外字 (12×12)
Gaiji16	外字 (16×16)
Gaiji20	外字 (20×20)

- 拡張子

拡張子	内容
*.ufh	for DT-900
*.ufv	for DT-930

フォント変換

作成または編集したユーザ定義フォントファイルを指定した機種用 (DT-900、DT-930、DT-930 エミュレータ) のユーザ定義フォントファイルまたは 1 文字ビットマップファイルに変換します。

DT-900、DT-930 用のユーザ定義フォントファイルは上記のファイル名と拡張子で出力します。
また、DT-930 エミュレータ用のユーザ定義フォントファイルは下記のファイル名で出力します。

- ユーザフォント定義ファイル (for DT-930 エミュレータ)

ファイル名	内容	タイプフェース名
UserAL6.ttf	縮小 ANK (6×6)	UserFontAL
UserAL6W.ttf	横倍縮小 ANK (6×6)	
UserAL8.ttf	縮小 ANK (8×8)	
UserAL8W.ttf	横倍縮小 ANK (8×8)	
UserAL10.ttf	縮小 ANK (10×10)	
UserAL10W.ttf	横倍縮小 ANK (10×10)	
UserAS6.ttf	標準 ANK (6×12)、漢字 (12×12)	UserFontAS
UserAS6W.ttf	横倍標準 ANK (6×12)、横倍漢字 (12×12)	
UserAS8.ttf	標準 ANK (8×16)、漢字 (16×16)	
UserAS8W.ttf	横倍標準 ANK (8×16)、横倍漢字 (16×16)	
UserAS10.ttf	標準 ANK (10×20)、漢字 (20×20)	
UserAS10W.ttf	横倍標準 ANK (10×20)、横倍漢字 (20×20)	
GaijiAS6.ttf	外字 (12×12)	GaijiFontAS
GaijiAS6W.ttf	横倍外字 (12×12)	
GaijiAS8.ttf	外字 (16×16)	
GaijiAS8W.ttf	横倍外字 (16×16)	
GaijiAS10.ttf	外字 (20×20)	
GaijiAS10W.ttf	横倍外字 (20×20)	

- 1 文字ビットマップファイル

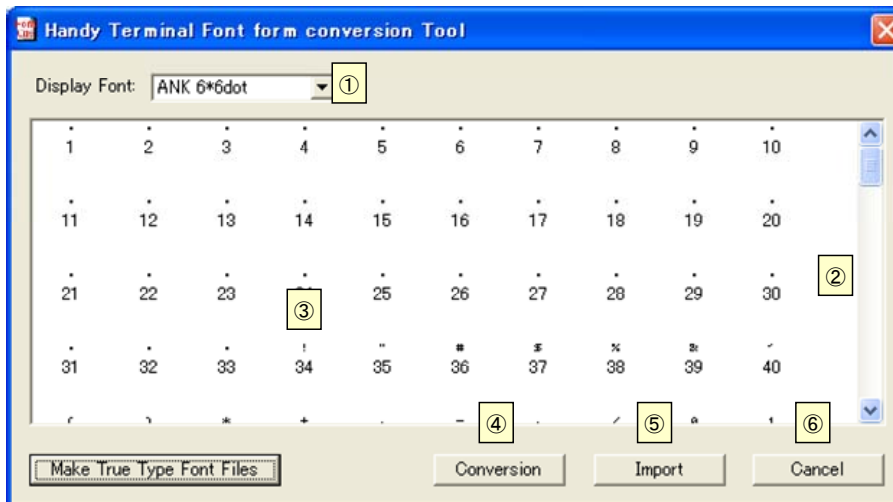
1 文字ビットマップファイルのファイル名は、フォントファイル名_ユーザ定義フォントファイル内のインデックス.bmp となります。

例) 標準 ANK (10×20) の「A」

ANK20_66.bmp

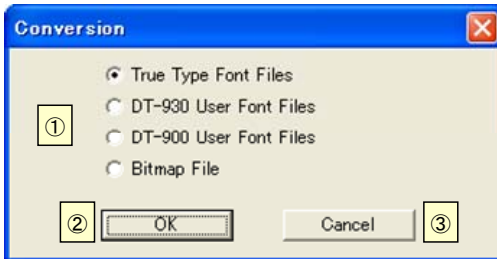
4.1.2. 画面構成

基本画面



	項目名	説明
①	Display Font	Font 表示領域に表示するフォント種類を指定します デフォルトは、ANK 6*6dot です フォント種類には、ANK6*6、ANK8*8、ANK10*10、ANK6*12、ANK8*16、ANK10*20、Kanji12*12-1、-2、-3、Kanji16*16-1、-2、-3、Kanji20*20-1、-2、-3、Gaiji12*12、Gaiji16*16、Gaiji20*20 があります
②	Font 表示領域	①で指定したフォントを表示する領域です 定義されていない文字は「・」で表示します
③	選択フォント	Font 表示領域に表示している 1 文字を選択することができます 選択すると他の文字と区別できるように、インデックスのバックグラウンドカラーが変化します
④	Conversion ボタン	Conversion 画面を表示します 詳細は下記の Conversion 画面を参照してください
⑤	Import ボタン	Import 画面を表示する 詳細は下記の Import 画面を参照してください
⑥	Cancel ボタン	フォントコンバータを終了します
⑦	Make ボタン	Import 画面を表示し、選択したフォントの True Type Font を生成します 詳細は下記の Import 画面を参照してください

Conversion 画面



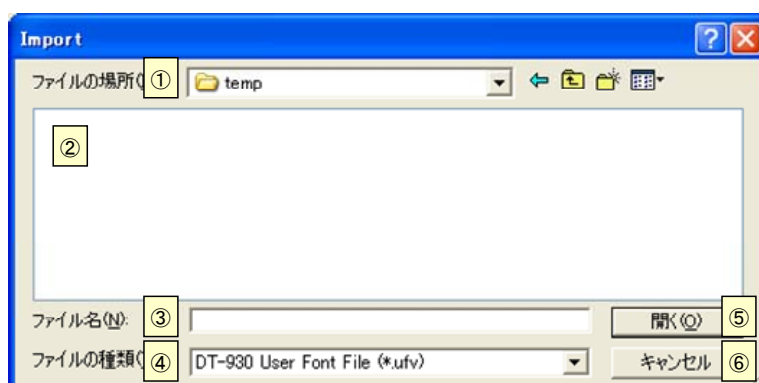
	項目名	説明
①	出力ファイルの種類	出力するファイル形式を選択します <ul style="list-style-type: none"> • True Type Font Files ユーザ定義フォントファイル (for DT-930 エミュレータ) • DT-930 User Font Files ユーザ定義フォントファイル (for DT-930) • DT-900 User Font Files ユーザ定義フォントファイル (for DT-900) • Bitmap File 1 文字ビットマップファイル 出力ファイルは「4.1.1 機能」に示したファイル名で保存します
②	OK ボタン	変換ファイルの保存フォルダを指定するダイアログを表示します (下記参照)
③	Cancel ボタン	Conversion 画面を閉じ、基本画面に戻ります



	項目名	説明
①	保存するフォルダ	ファイルを保存するフォルダを指定します
②	OK ボタン	ファイルを作成し保存します
③	キャンセル ボタン	本ダイアログを閉じ、Conversion 画面に戻ります

※ 本ダイアログで指定したフォルダ内に、出力ファイルと同名のファイルが存在する場合は、確認メッセージを出力します。詳細は下記のメッセージダイアログを参照してください。

Import 画面



	項目名	説明
①	ファイルの場所(I)	取り込むファイルが存在するフォルダを選択します
②	ファイル表示領域	①で指定したフォルダ内にある④で指定した拡張子のファイルを表示します
③	ファイル名(N)	取り込むファイルを指定します(複数指定可能) 「4.1.1 機能」に示したすべてのファイルを指定しなくても、エラーにはなりません 指定したファイルに関するフォントのみ、基本画面の Font 表示領域を変更します 「4.1.1 機能」に示したファイル名と異なるファイルを指定した場合はエラーメッセージを出力し、取り込みを停止します
④	ファイルの種類(I)	取り込むファイルの種類を拡張子で指定します
⑤	開く(O) ボタン	Import ボタンから呼び出した場合： ③で指定したファイルをもとに、基本画面の Font 表示領域を更新し、基本画面に戻ります Make ボタンから呼び出した場合： ③で指定したファイルをもとに、True Type Font を生成し、基本画面に戻ります
⑥	キャンセル ボタン	Import 画面を閉じ、基本画面に戻ります

メッセージダイアログ

- エラーメッセージダイアログ

本ツール使用時に、エラーが発生した場合は、エラーメッセージダイアログを表示します。

図中の点線内に下表のメッセージのいずれかを表示します。



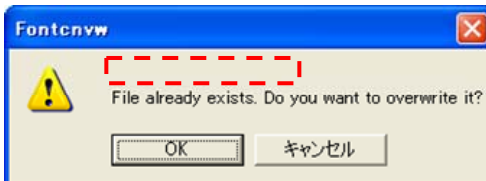
エラーメッセージ	説明
The KCG file is not found. Application will close.	DT-900.kcg ファイルが存在しない場合
The memory which this application uses is short.	メモリが不足している場合
The formats of the KCG file are different. Application will close.	DT-900.kcg のバージョンが異なる場合
File name is different.	*.ufv、*.ufh のファイル名が規定と異なる場合
Please choose 1 character to change.	ビットマップファイルをインポートするときに変更する文字を選択していない場合
Size of 1 character to change is different from a Bitmap file.	ビットマップファイルをインポートするときを選択した文字とビットマップファイルのサイズが異なる場合
The BIN files are not found.	TrueTypeFont ファイルの入力となる BIN ファイルが存在しない場合
This application cannot read an external character file.	外字ファイルを読み込めない場合

- 確認メッセージダイアログ

ユーザフォント定義ファイル保存時に、指定したフォルダ内に同名のファイルが存在する場合は、確認メッセージダイアログを表示します。

ファイルを上書きする場合は「OK」ボタンを、上書きしない場合は「キャンセル」ボタンを押下してください。

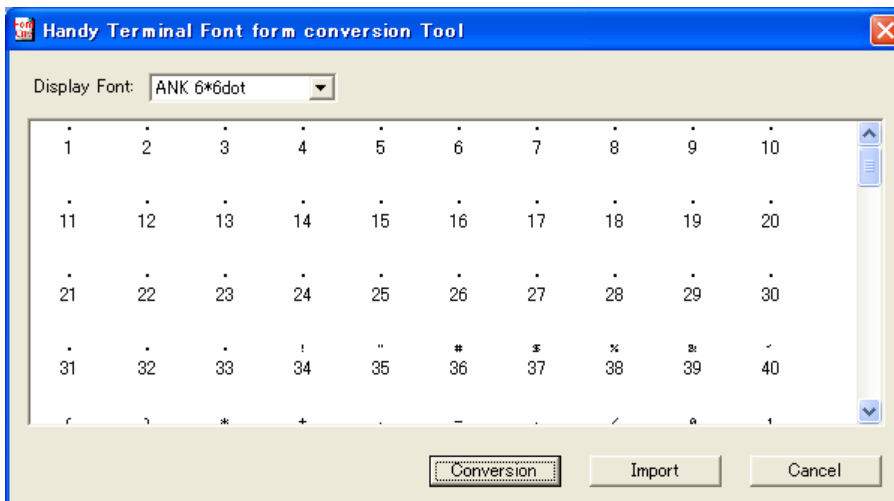
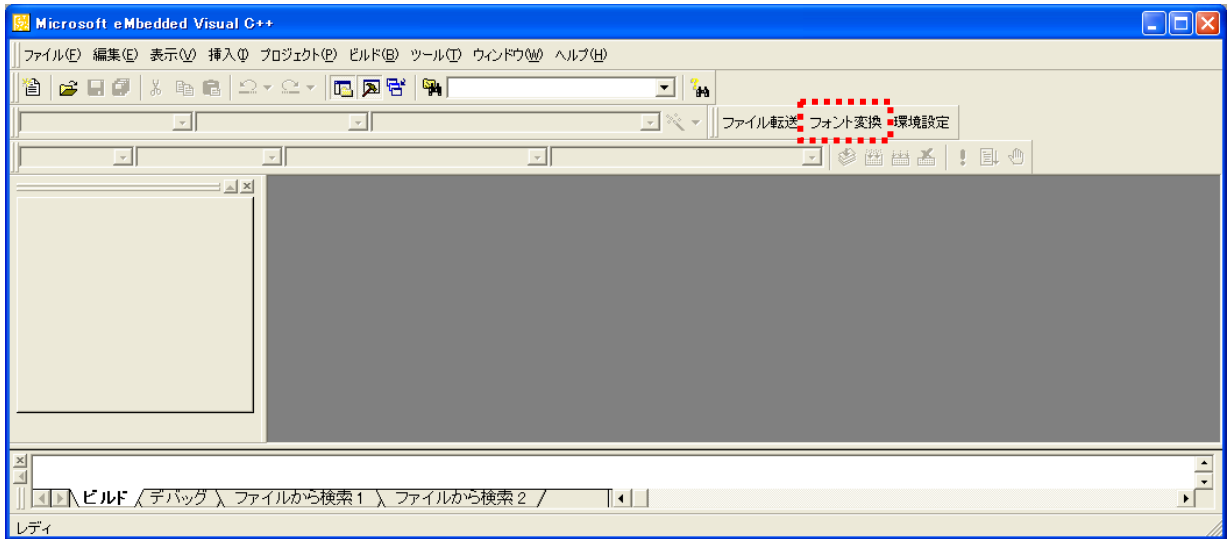
図中の点線内に対象のファイル名を表示します。



4.1.3. 使用方法

(1) 起動方法

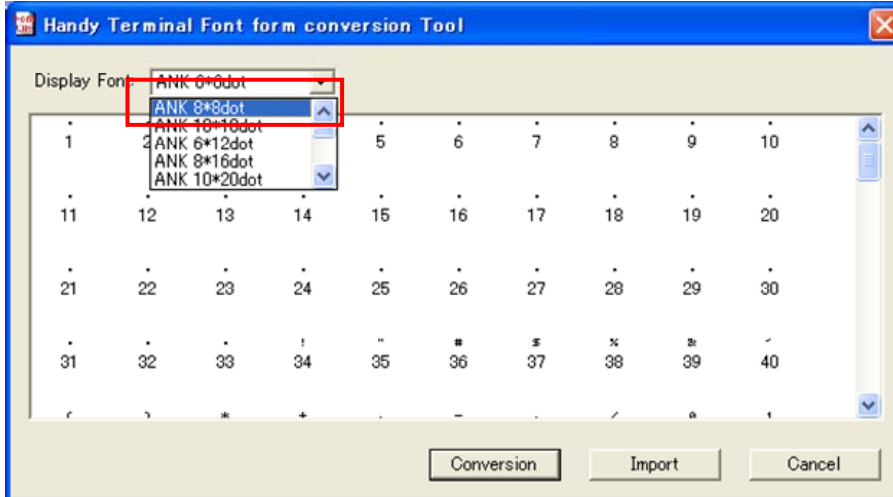
ツールバーの「**フォント変換**」をクリックすることにより、「4.1.2 画面構成」に示した基本画面が起動します。



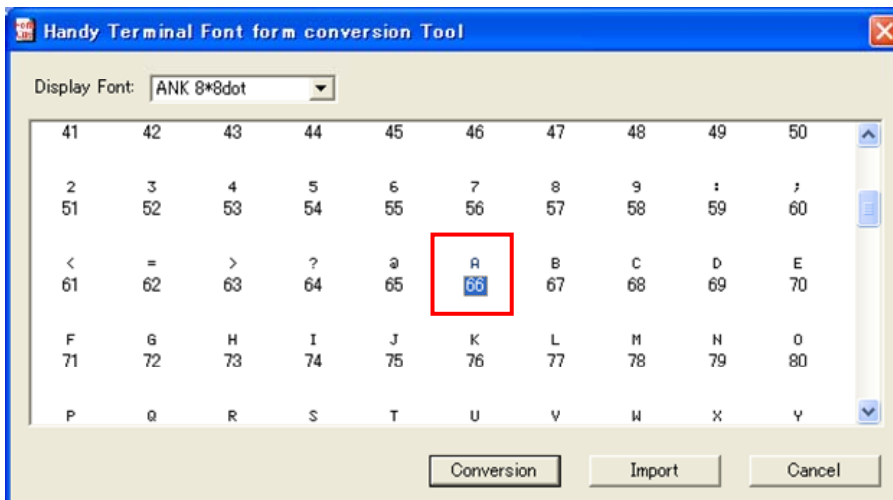
(2) ユーザ定義フォントファイルの新規作成

ユーザ定義フォントファイルを新規作成します。ここでは、縮小 ANK (8×8) の「A」を変更し、ユーザ定義フォントファイル (for DT-930 エミュレータ) を作成する手順を示します。

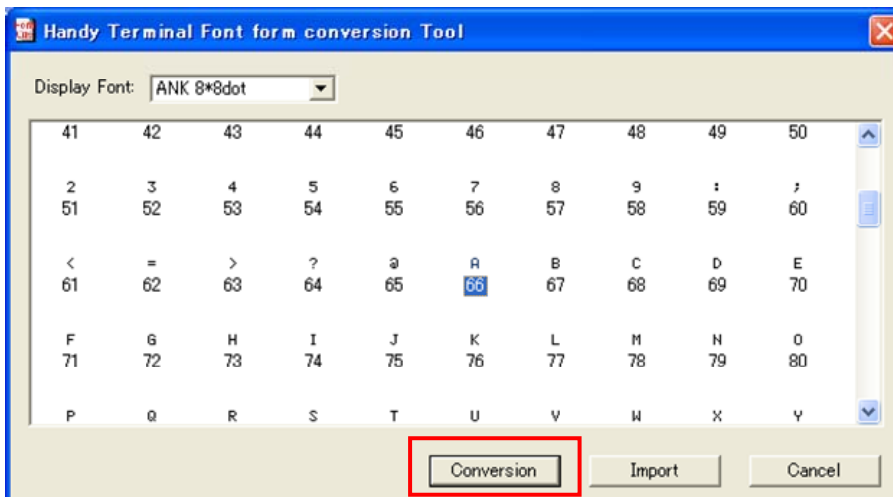
1. フォントコンバータを起動し、Display Font を「ANK 8*8dot」に変更します。



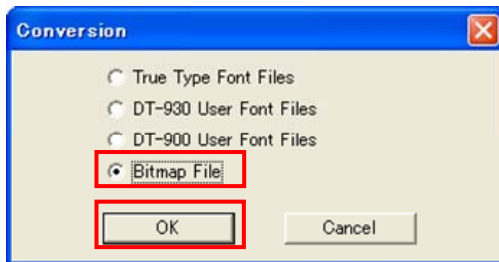
2. Font 表示領域に表示している「A」を左クリックし、インデックスナンバーのバックグラウンドカラーが変化したことを確認します。



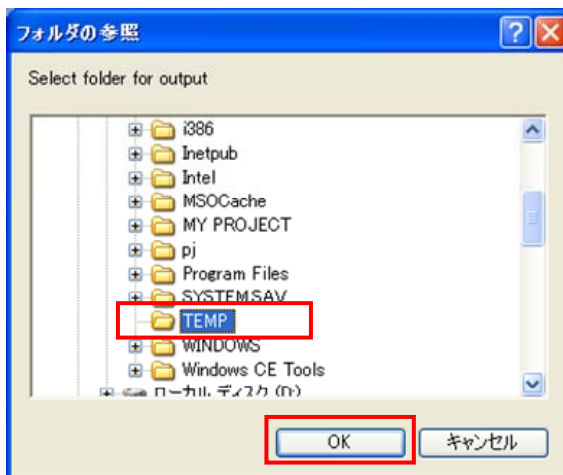
3. 「Conversion」ボタンを押下し、Conversion 画面が起動したことを確認します。



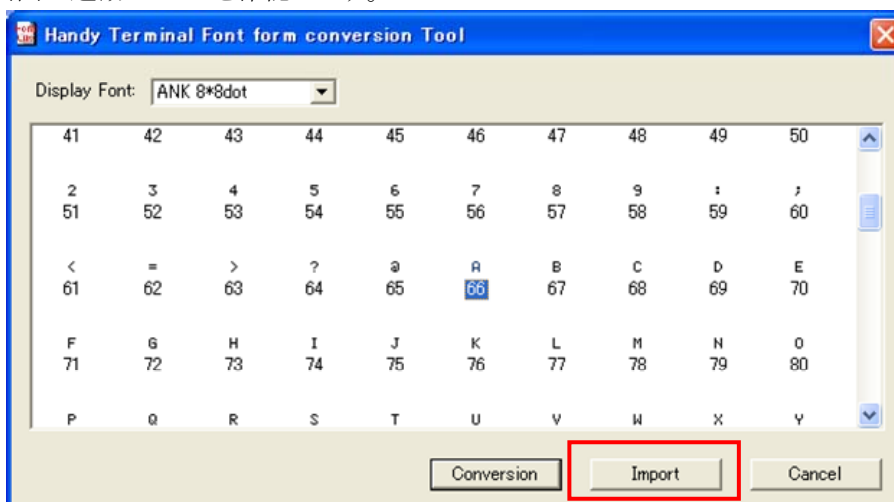
4. 「Bitmap File」にチェックを入れ、「OK」を押下します。



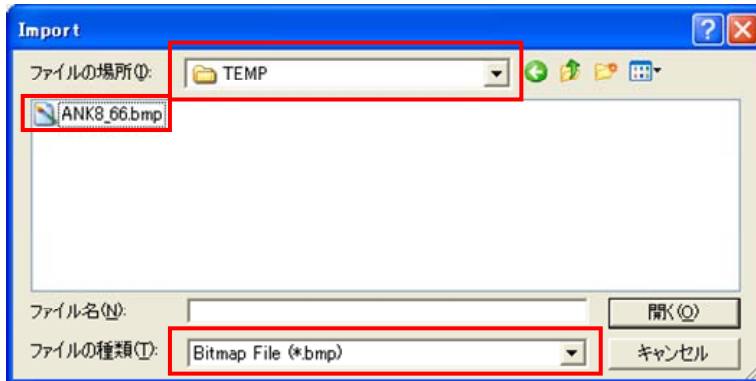
5. 出力する任意のフォルダを選択し、「OK」を押下します。ここでは、C:¥TEMP に出力します。出力フォルダ内に同名のファイルが存在する場合は上書き確認ダイアログを表示します。上書きしない場合は別のフォルダを指定してください。



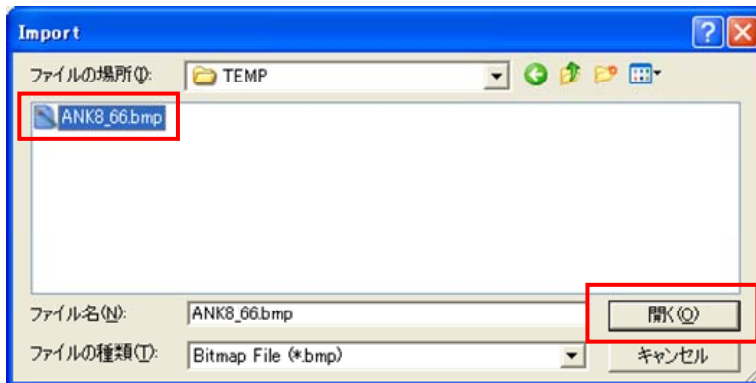
6. C:¥TEMP に ANK8_66.bmp が存在することを確認し、ビットマップエディタを用いて編集し、ANK8_66.bmp を上書き保存します。
7. 手順 1.および 2.と同様に、「ANK 8*8dot」の「A」を選択した状態で、「Import」ボタンを押下し、Import 画面が起動したことを確認します。



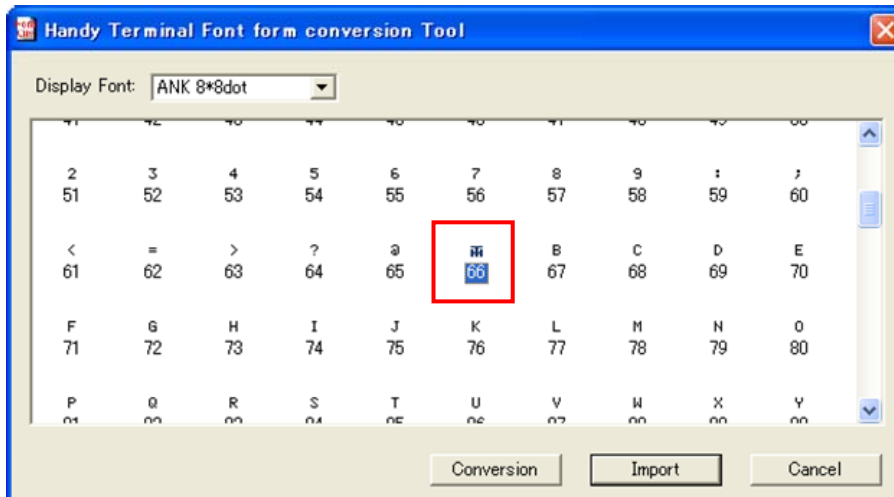
8. 「ファイルの場所(I)」に手順 5.で指定した C:¥TEMP を、「ファイルの種類(T)」に「Bitmap File (*.bmp)」を選択すると、手順 6.で編集した ANK8_66.bmp が表示されていることを確認します。



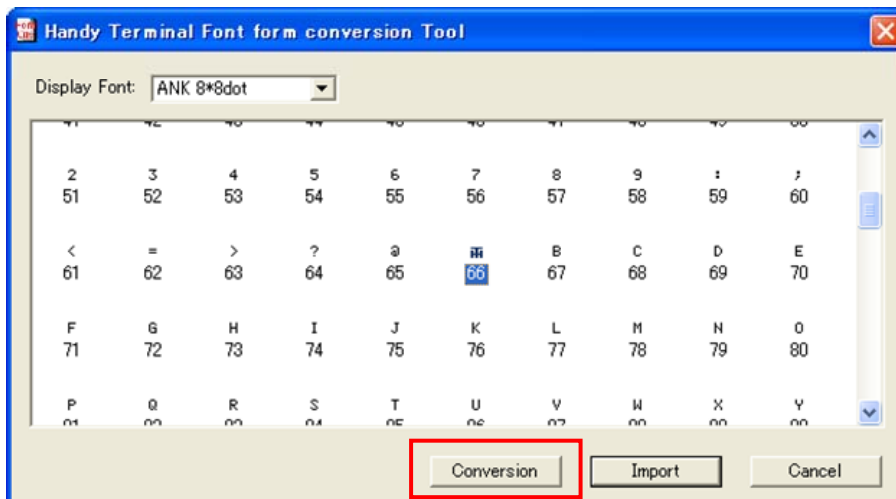
9. ANK8_66.bmp を選択し、「開く(O)」ボタンを押下します。



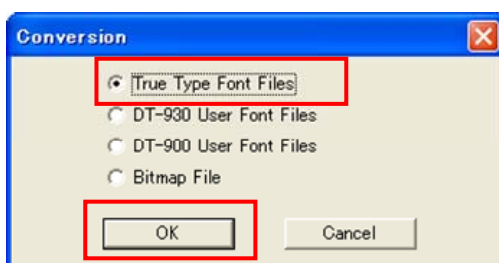
10. Font 表示領域内の「A」が手順 6.で編集した文字に変わっていることを確認します。



11. 「Conversion」ボタンを押下し、Conversion 画面を起動したことを確認します。



12. 「True Type Font Files」にチェックを入れ、「OK」ボタンを押下します。



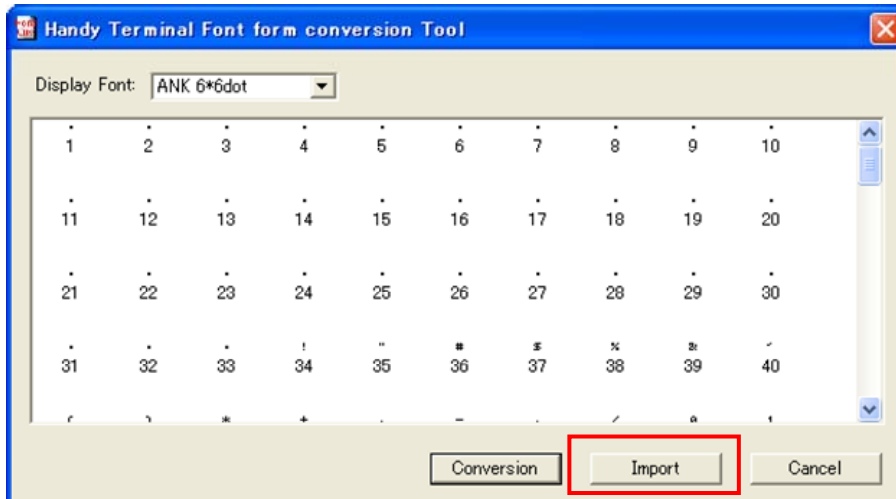
13. 手順 5.と同様に、出力先フォルダに C:¥TEMP を選択し、「OK」ボタンを押下します。
出力フォルダ内に同名のファイルが存在する場合は上書き確認ダイアログを表示します。上書きしない場合は別のフォルダを指定してください。
14. C:¥TEMP フォルダ内に「4.1.1 機能」のユーザフォント定義ファイル (for DT-930 エミュレータ) に示したファイルが存在することを確認します。

ユーザ定義フォントファイル (for DT-900、DT-930) を新規作成する場合は、手順 12. で「DT-900 User Font Files」または「DT-930 User Font Files」を選択してください。

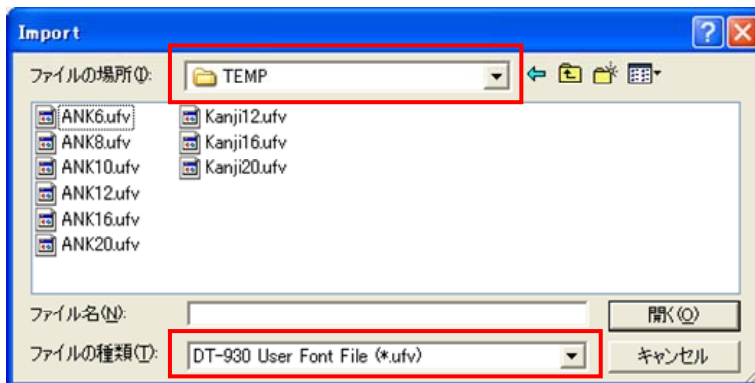
(3) 既存のユーザ定義フォントファイルの編集

既存のユーザ定義フォントファイル(for DT-930)を編集します。ここで使用する既存のユーザ定義フォントファイルは、漢字(20×20)の「亜」を「DT930」に変更しているものとします。ここでは、「DT930」を変更し、ユーザ定義フォントファイル(for DT-930)を編集する手順を示します。

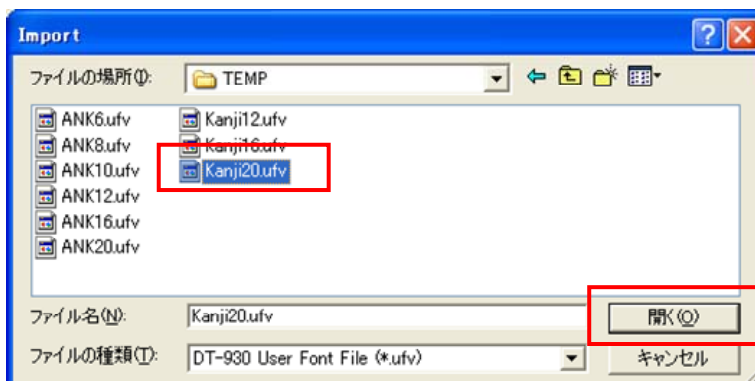
1. 既存のユーザ定義フォントファイルを「4.1.1 機能」に示したファイル名および拡張子に変更します。
2. フォントコンバータを起動し、「Import」ボタンを押下します。



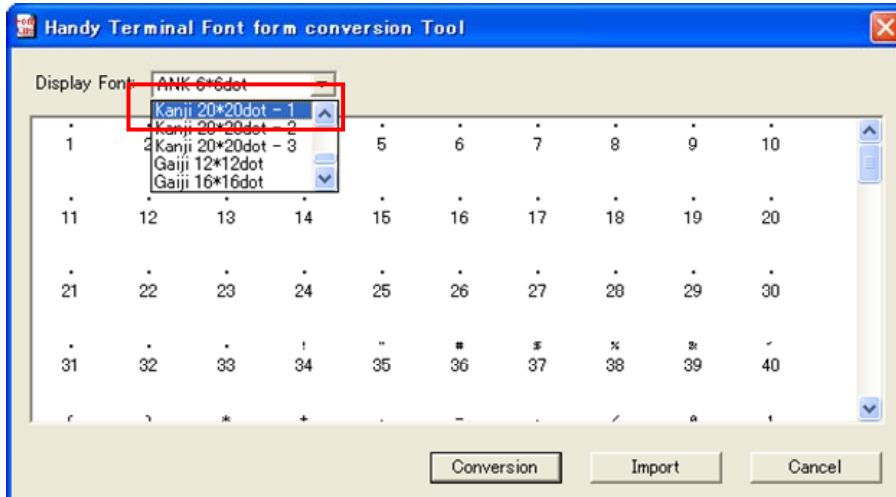
3. 「ファイルの場所(I)」に手順 1.でファイル名および拡張子を変更したファイルが存在するフォルダを、「ファイルの種類(T)」に「DT-930 User Font File (*.ufv)」を選択します。ここでは、C:¥TEMP にファイルがあるものとします。



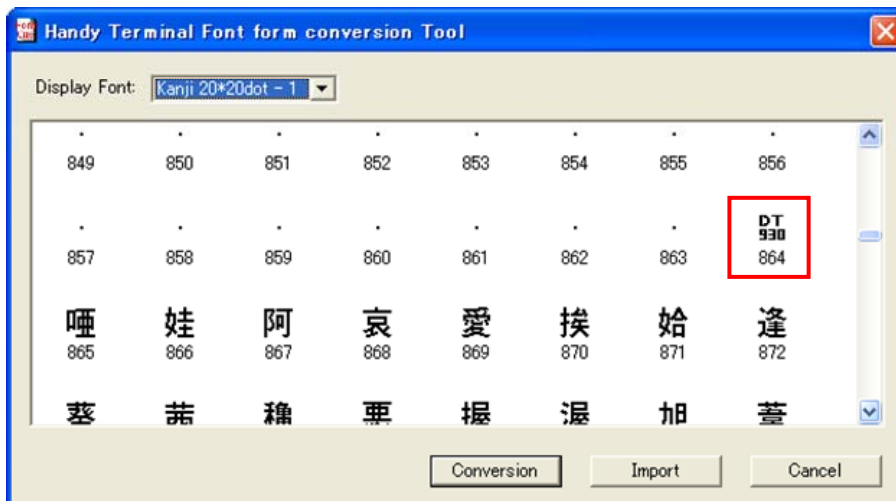
4. 「Kanji20.ufv」を選択し、「開く(O)」ボタンを押下します。
(ファイルを複数選択することも可能です。ここでは漢字(20×20)を編集するため、Kanji20.ufv のみを選択しています。)



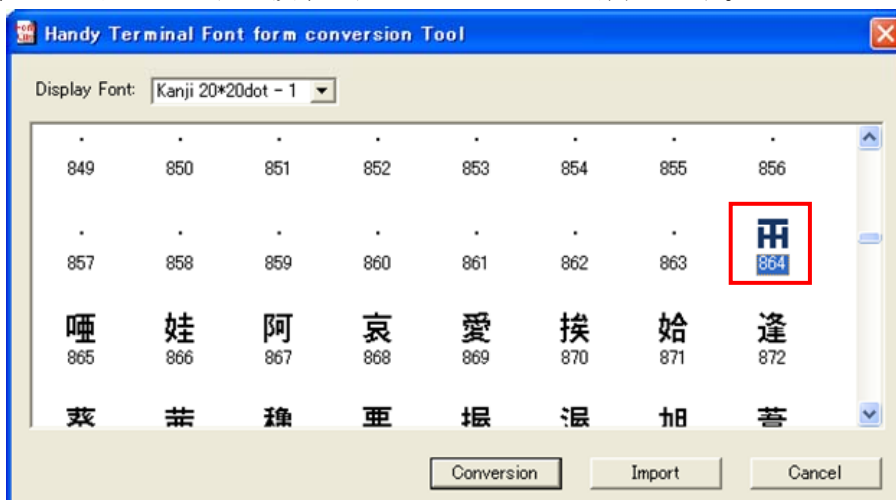
5. Display Font を「Kanji 20*20dot - 1」に変更します。



6. 「亜」が「DT930」に変わっていることを確認します。



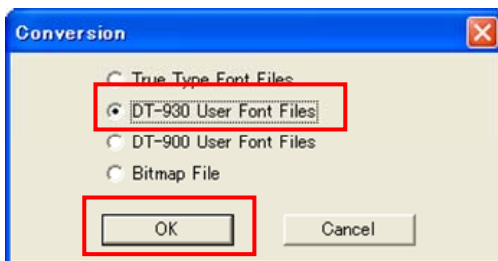
7. 「(2) ユーザ定義フォントファイルの新規作成」の手順 3.~10.と同様に、「DT930」を編集し、その編集した文字が Font 表示領域に表示されていることを確認します。



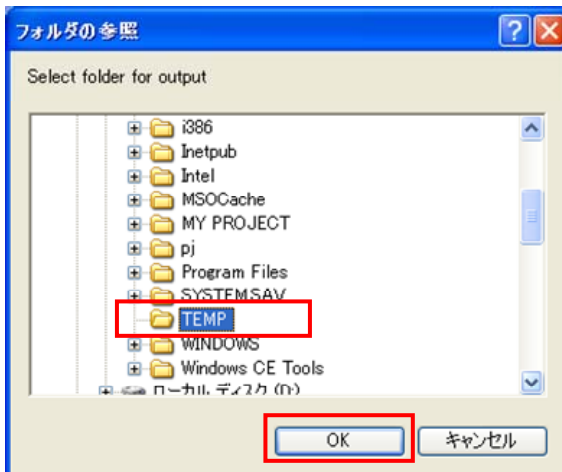
8. 「Conversion」ボタンを押下し、Conversion 画面を起動したことを確認します。



9. 「DT-930 User Font Files」にチェックを入れ、「OK」ボタンを押下します。



10. 出力する任意のフォルダを選択し、「OK」を押下します。ここでは、C:¥TEMP に出力します。出力フォルダ内に同名のファイルが存在する場合は上書き確認ダイアログを表示します。上書きしない場合は別のフォルダを指定してください。



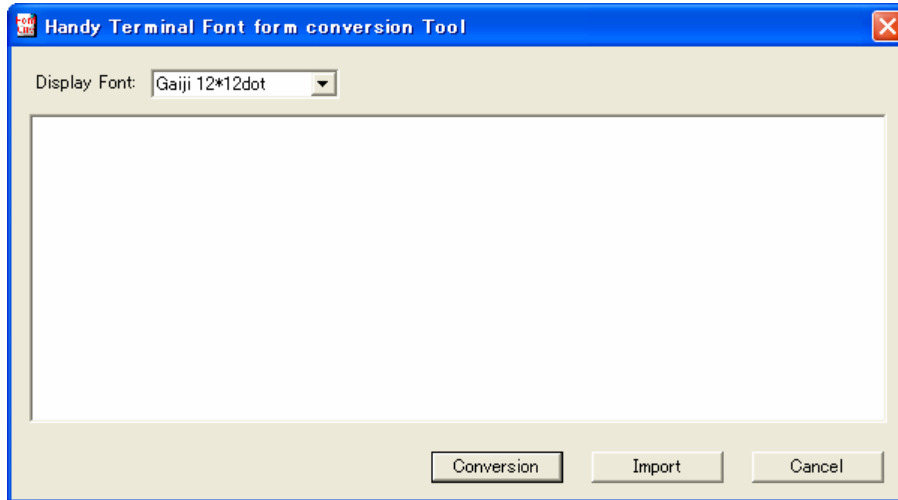
11. C:¥TEMP フォルダ内に「4.1.1 機能」のユーザフォント定義ファイル (for DT-930) に示したファイルが存在することを確認します。

編集したユーザ定義フォントファイル (for DT-930) を DT-930 エミュレータ、DT-900 で使用する場合は、手順 9. で「True Type Font Files」または「DT-900 User Font Files」を選択してください。また、手順 3. で「ファイルの種類(T)」に「DT-900 User Font File (*.uff)」を選択することにより、DT-900 用のユーザ定義フォントファイルの編集が可能となります。

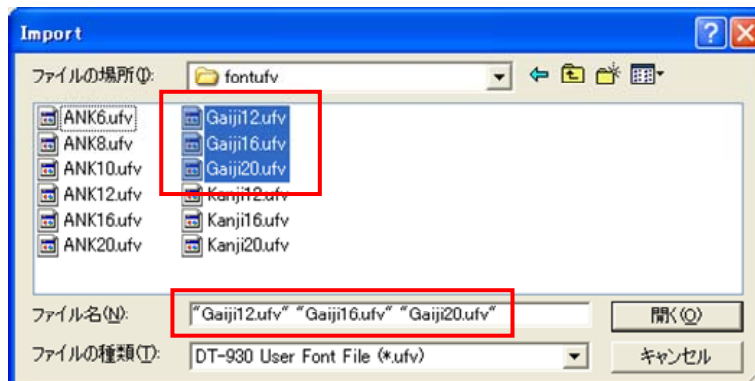
4.1.4. 注意事項

外字について

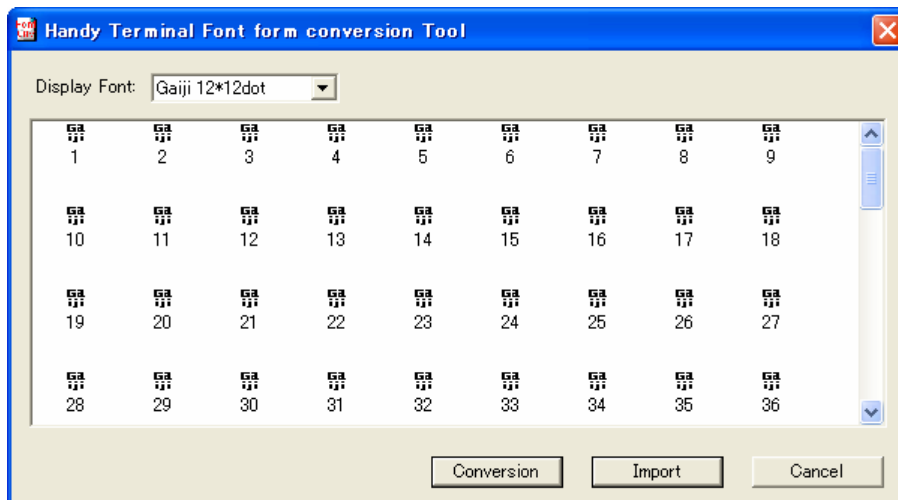
本ツールの初回起動時は、Display Font を「Gaiji 12*12dot」、「Gaiji 16*16dot」、「Gaiji 20*20dot」に設定しても、Font 表示領域に何も表示しません。



外字を編集するためには、「4.1.3 使用方法 (3) 既存のユーザ定義フォントファイルの編集」の手順 3.と同様の手順で既存の外字ファイルを取り込む必要があります。



取り込んだ外字ファイルが Font 表示領域に反映されていることを確認してから、編集を行ってください。



4.2. LMWIN

4.2.1. 機能

アップダウンロードユーティリティソフトウェアは、以下の機能を実行することができます。

1. ホスト PC と HT 間のファイルの送信、受信
2. 各種コマンドの実行(ブザー鳴動、フォーマット)
3. ホスト PC 上でのスクリプトファイル実行
4. USB または RS-232C 接続による通信機能

また、ホスト PC と HT 間の以下にあげる 15 のコマンド機能があります。

1. ファイル送信
2. ファイル受信
3. ファイル追加送信
4. ファイル追加受信
5. ブザー鳴動
6. フォーマット
7. 日時設定
8. 削除
9. 移動/名前変更
10. ディスク情報取得
11. ファイル情報取得
12. ファイル情報設定
13. セッション終了
14. 子プロセス実行
15. 文字列表示

4.2.2. 使用方法

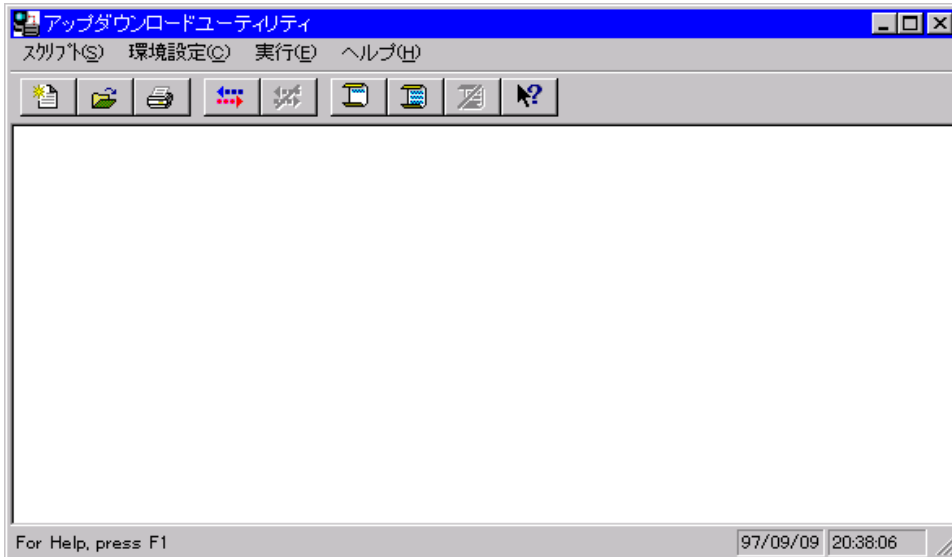
(1) 起動方法

インストールしたディレクトリ内の LMWIN32.EXE を実行してください。

(2) メインメニュー

以下の画面を表示します。4つのメニューがメニューバーにあります。

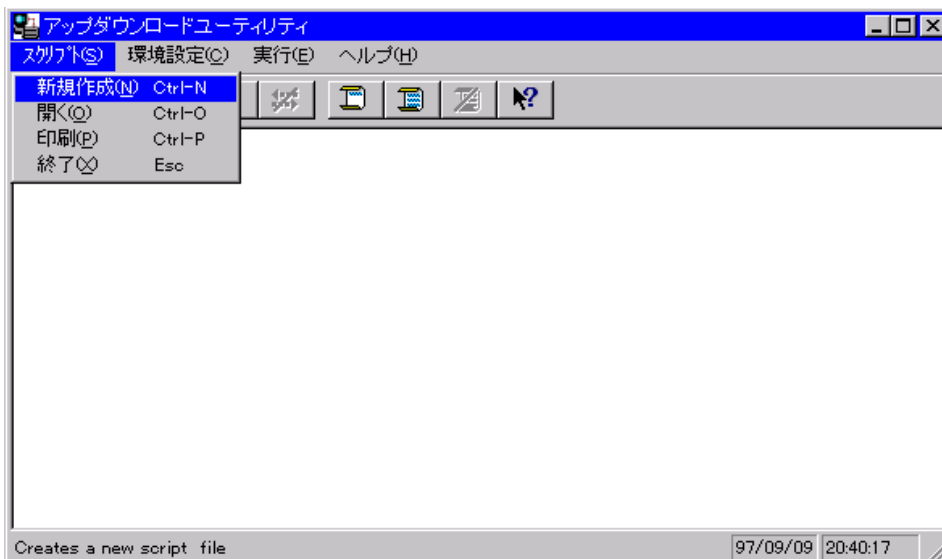
- スクリプト :スクリプトメニューが表示されます
- 環境設定 :環境設定メニューが表示されます
- 実行 :通信メニューが表示されます
- ヘルプ :ヘルプ画面へ



(3) スクリプトメニュー

スクリプトメニューには下記のプルダウンサブメニューがあります。

- 新規作成 :新しいスクリプトファイルを作成します。
- 開く :既存のスクリプトファイルを開き、編集します。
- 印刷 :スクリプトファイルを印刷。現在は実行できません。
- 終了 :アップダウンロードユーティリティを終了します。



(4) スクリプトファイル新規作成

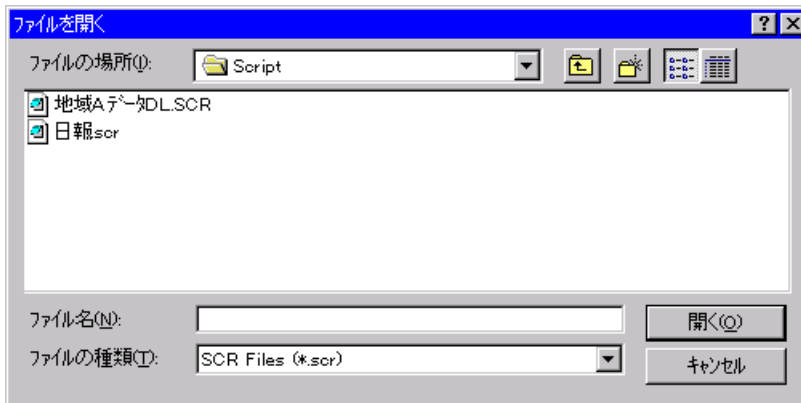
「新規作成」サブメニューオプションを選択すると、下の画面になります。

- 追加 : コマンドを「スクリプト」に追加します
- 結合 : 前のスクリプトコマンドに結合します
- 戻る : メイン画面に戻ります スクリプトファイルのセーブはしません
- 保存 : スクリプトファイルを上書き保存します
- 名前を付けて保存 : 名前を付けて保存します
- 参照 : HT に送るファイルを選択できます



(5) スクリプトファイルを開く

スクリプト選択画面でスクリプトファイルを選択すると次の画面になります。



スクリプト選択画面でスクリプトファイルを選択すると次の画面になります。

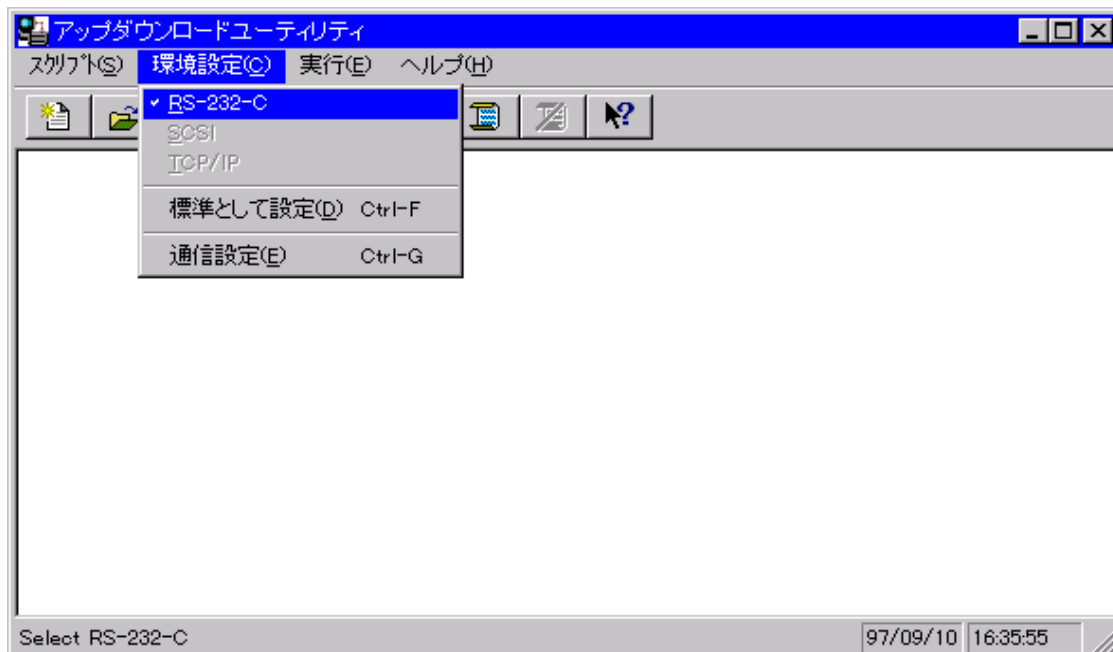


「スクリプトファイル」に選択したスクリプトファイルの内容が記述されます。

(6) 環境設定メニュー

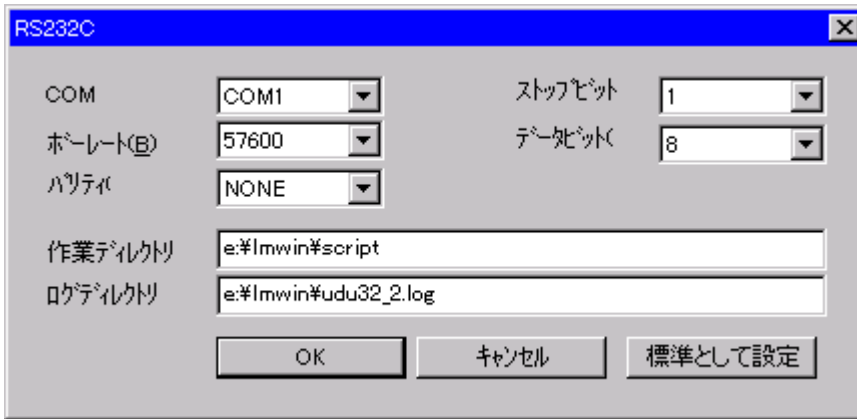
環境設定メニューには以下のプルダウンサブメニューがあります。

- RS-232C : RS-232C で通信します
- SCSI : SCSI で通信します
- TCP/IP : TCP/IP で通信します
- TCP/IP (New) : TCP/IP で通信します
- B.S/B.B (IO BOX) : B.S/B.B I/O BOX を介して通信します



ここでは、例として次のように設定しています。

COM ポート : COM1
ボーレート : 57600BPS
パリティ : なし
ストップビット : 1 ビット
データ長 : 8 ビット



(8) 実行

実行メニューには下記のプルダウンサブメニューがあります。

「**接続**」→「**開始**」サブメニューは HT からのコマンド受け付け状態になります。

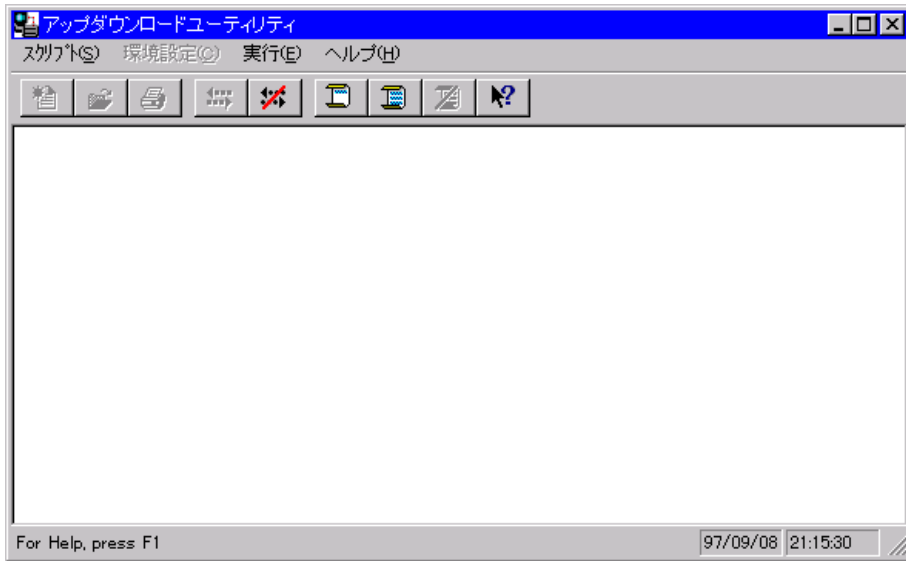
「**コマンド**」サブメニューはコマンド実行画面を表示します。

「**スクリプト**」サブメニューは実行スクリプト選択画面を表示します。

「**中断**」サブメニューは通信中に選択でき、通信を中断します。



サーバーモードを開始することにより通信を初期化して、HT から来る接続要求の受け付け状態へアップダウンロードユーティリティを移行させます。



(9) LMWIN コマンド

コマンド画面では 1 種類のコマンドを実行することができます。

コマンド実行時はモード設定を行なうことができ、1 回モードが選択されると、HT との通信接続は切り離しません。

また、連続モードが選択されると、「**中断**」サブメニューが選択されるまで繰り返し実行します。デフォルトは連続モードです。



(10) スクリプト実行

スクリプト選択画面では1種類のスクリプトを選択できます。

選択したスクリプトを実行します。



(11) インストール

DT-930をI/Oボックスに載せて、次の手順でDT-930本体側のインストール準備を行い、インストールを開始します。

1. 「**S**」キーと「**.**」キーを押しながら電源をオンします(システムメニュー起動)。
2. 「**4**」キーを押します(「**4:テンソウ**」メニューが起動)。
3. 「**5:プロトコル**」を設定します。
FLINKを選択します。
4. 「**6:ポート**」を設定します。
I/Oボックス経由に設定します。
5. 「**7:ソクド**」を設定します。
PC側、I/Oボックスの設定に合わせて通信速度を設定します。
6. 「**4:ユーティリティ**」を選択します。
「**4**」キーを押すと「ユーティリティ」メニューが表示されます。
7. 「**1:ファイルジュシシ**」を選択します。

以上で通信待機状態になります。

PC側が起動していれば通信が開始されます。

ダウンロード完了後は「**CLR**」キーを押して、「システムメニュー」のトップページに戻り、「**1:APキドウ**」を行うと、アプリケーションプログラムが起動します。

4.3. MultiDrop

4.3.1. 機能

アップダウンロードシステムは、HT 本体とホスト PC の間で以下の業務を行えることを目的としています。

アプリケーションインストール	:ホスト PC から HT 本体にアプリケーションをインストール
ファイル送信	:ホスト PC 上のファイル(データ)を HT 本体に送信
ファイル受信	:HT 本体上のファイル(データ)をホスト PC にて受信

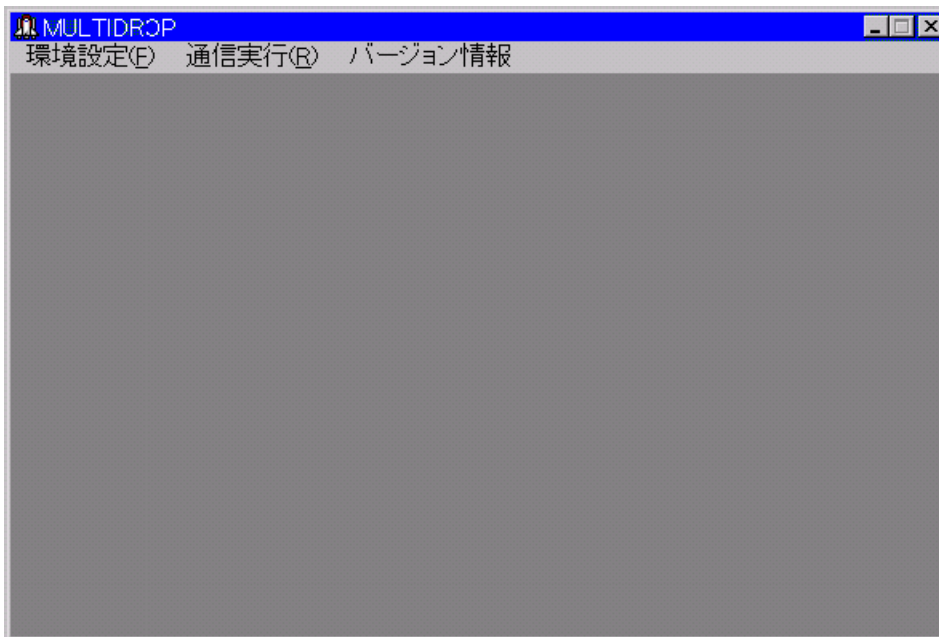
4.3.2. 使用方法

(1) 起動方法

インストールしたディレクトリ内の MULTI32J.EXE を実行してください。

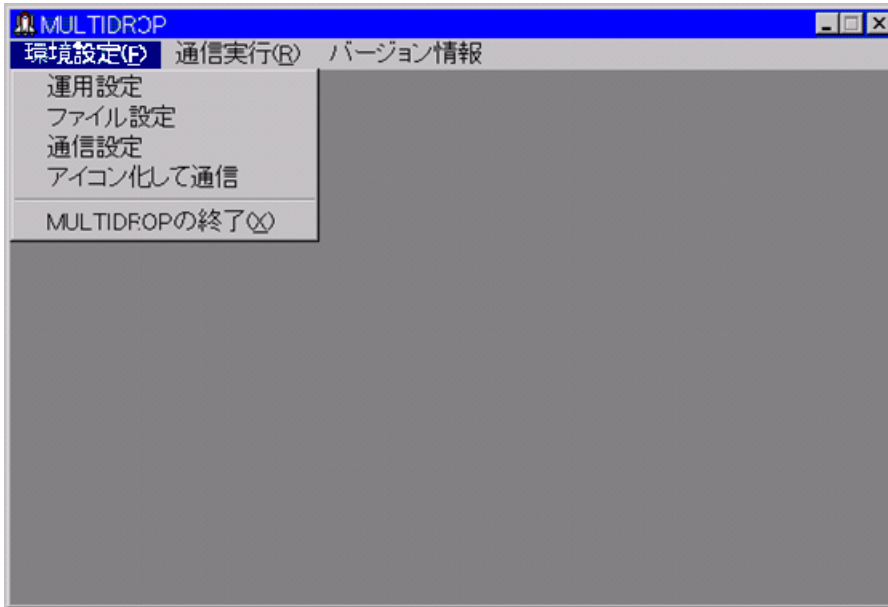
(2) 基本画面

基本画面は、ホストユーティリティ起動時に最初に表示される画面です。
環境設定、通信実行、バージョン表示を選択します。



(2) 環境設定画面

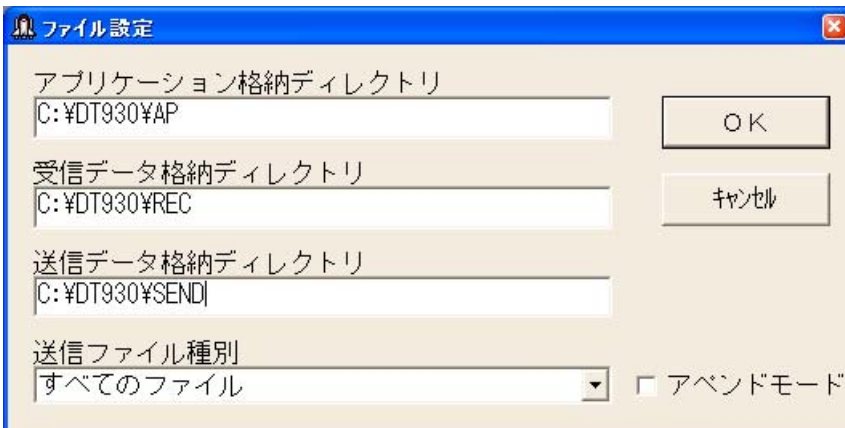
環境設定画面はメニューバーの環境設定をクリックすることにより表示します。



(3) ファイル設定画面

ファイル設定メニューは、環境設定メニューでファイル設定を選択することにより表示します。

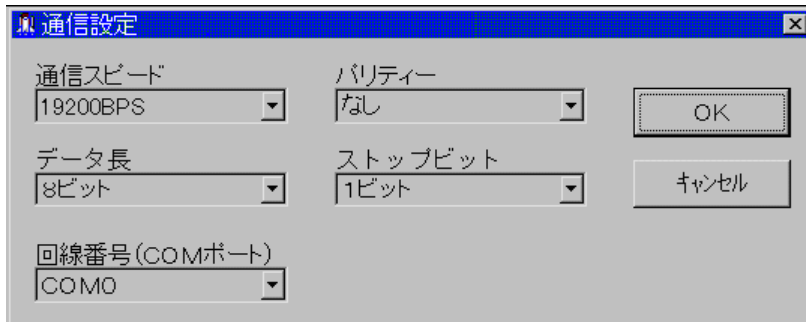
この画面でファイル格納ディレクトリの設定、送信ファイル種別の設定値の確認および変更、およびアペンドモードの設定を行います。



(4) 通信設定画面

通信設定メニューは環境設定メニューで通信設定を選択することにより表示します。
この画面で通信速度、データ長、パリティ、ストップビットの設定値の確認、および変更を行います。
通信速度を以下の5つより選択します。

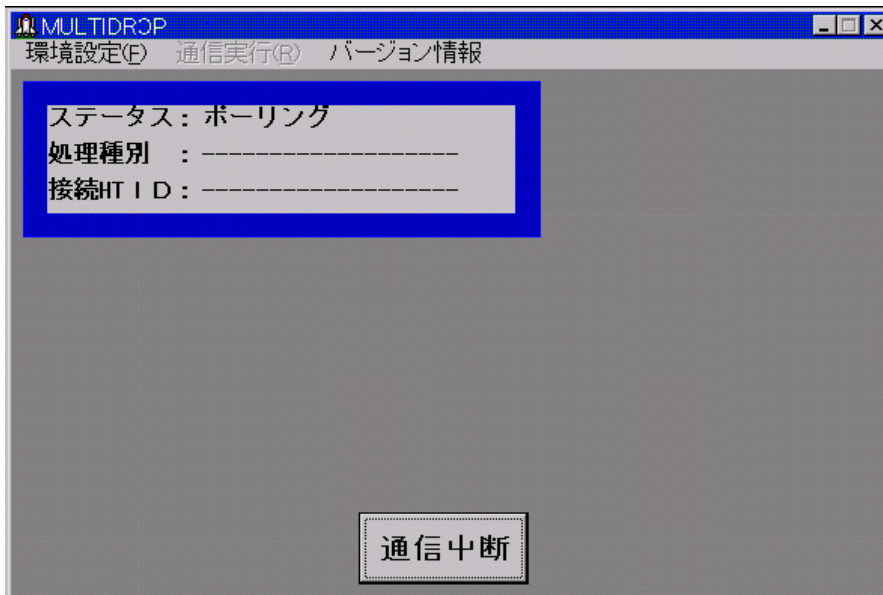
- 1200BPS
- 2400BPS
- 4800BPS
- 9600BPS
- 19200BPS



(5) 通信実行画面

通信実行画面は、基本画面で通信実行を選択することにより表示する通信実行中画面です。
通信実行中は、その処理フェイズによって、ポーリング中と通信中の2種類があります。

- ポーリング中の画面



- 通信中の画面



(6) システムメニュー画面

HT ユーティリティを操作するには、最初にシステムメニューを起動しなければなりません。システムメニューが起動すると、次のようなメニューを表示します。

メニュー
1:APキドウ
2:モードセット
3:ヒヅケ/ジコク
4:テンソウ
5:バージョン

「4:テンソウ」を選択してください。

(7) 転送メニュー画面

テンソウ
1:ドゥホウインストール
2:APインストール
3:コキサクセイ
4:ユーティリティ
6:ポート SIR
7:ソク 115k

「2:AP インストール」を選択してください。

カシオ計算機お問い合わせ窓口

製品に関する最新情報

- 製品サポートサイト（カシオペア・ハンディターミナル）

<http://casio.jp/support/pa/>

製品の取扱い方法のお問い合わせ

- 情報機器コールセンター



0570-022066

市内通話料金でご利用いただけます。

携帯電話・PHS 等をご利用の場合、**048-233-7241**

カシオ計算機株式会社

〒151-8543 東京都渋谷区本町 1-6-2

TEL 03-5334-4638(代)